REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
07-July-95

3. REPORT TYPE AND DATES COVERED
Final Report, 1 Oct. 94 - 29 May 95

4. TITLE AND SUBTITLE
Design for Silicon Cortex

5. FUNDING NUMBERS
(G) DAAH04-94-C-0072

6. AUTHOR(S)
Carl F. R. Weiman and Andrew F. Laine

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Transitions Research Corporation
Shelter Rock Lane
Danbury, CT 06810-8159

8. PERFORMING ORGANIZATION REPORT NUMBER
TRC07JUL

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This research analyzed mathematical properties of the discrete Gabor filter and tested it on real imagery. Several unexpected results emerged. The discrete Gabor filter, is not a wavelet, and is not a complete image decomposition function. To complete the set, additional filters are required.

The Gabor filter measured binocular disparity to 1/100th pixel accuracy. However, there was an large phase spread across contrast boundaries, due to the poor bandpass of analog video signals feeding the frame buffer. In recognizing handwritten characters, the discrete Gabor filter performed more poorly than wavelets, because of the lack of components which would have recognized junction topologies of characters with "X" and "Y" subgeometries. This failure is related to the lack of completeness discovered in the mathematical analysis.

Such negative results preclude the design of a general purpose "silicon cortex" architecture. Nevertheless, later studies could augment the components proposed here, testing them on a high performance general purpose processor such as the C80 system from GIC, or the CNAPS.

14. SUBJECT TERMS
robot vision, stereo, Gabor function, phase disparity, wavelet

15. NUMBER OF PAGES
53

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

REF D

DESIGN FOR SILICON CORTEX

Final Report

BMDO STTR Phase I Contract # DAAH04-94-C-0072

June 1995

Carl F. R. Weiman, Ph. D.,
Principal Investigator

Transitions Research Corporation
Shelter Rock Lane
Danbury, CT., 06810-8159
(203) 798-8988
weiman@trc.com

and

Andrew F. Laine, Ph. D.
co-Investigator

Department of Computer and Information Sciences
Computer Science and Engineering Building, E324
University of Florida
Gainesville, FL 32611-2024
(904) 392-1239
laine@cis.UFL.edu

TABLE OF CONTENTS

1. BACKGROUND

1.1 Overview of Objectives and Results

The objective of this study was to analyze the complexity and demonstrate the performance of an efficient discrete Gabor filter which could be used as a building block in a highly parallel image processing architecture. Applications include binocular stereo vision for real-time robots, pattern recognition, and the entire gamut of image processing.

Artificial vision algorithms are outstripping the ability of computation platforms to execute them in timely fashion. "Silicon Retinas" alleviate some of the low-level computation by parallelism, and the replacement of digital computations by analog ones. The role of the proposed discrete Gabor filter is a "Silicon Cortex" which performs the next higher levels of image processing, such as binocular disparity and spatial frequency analysis, using massive fine-grained parallelism of universal computation elements.

Questions which we answered during the study fell into the following categories:

1) Is the discrete Gabor filter a universal computation element?
2) Is the discrete Gabor filter computationally more efficient than alternatives?
3) Does the discrete Gabor filter perform well on real imagery?

In special cases, the answers to these questions were a qualified "yes". For the first question, the answer was generally no because the filter is not a wavelet, is not complete, and is missing some low-order derivatives. The latter could be augmented *ad hoc*, a topic for later study.

For the second question, the answer is negative; the difference-of-Gaussians requires fewer operations for general image decomposition. But in cases where the information relevant to the task narrows to a single orientation channel, for example, binocular disparity, or Lie group transforms in general (rotations, zooms), the discrete Gabor *is* more efficient.

For the third question, the answer is positive for binocular stereo, but negative for character recognition. The success of the former derives from the efficiency of the filter is sorting geometric information channels. The failure of the latter is due to the absence of computational universality, specifically, the lack of second order cross-derivatives which recognize local topological junctions such as are present in the letters "Y" and "X".

Below we describe the discrete Gabor filter in sufficient detail for discussion of the results of the research which follows.

1.2 The Discrete Gabor Filter

Since Gabor's introduction of the function now bearing his name in 1946 (See [11] in Section 6, References), the definition has been extended to two dimensions and proposed as a model for the simple and complex cells of the human visual cortex by Daugman [6, 7] and others. The advantageous image processing features of the function include efficient separation of image information into geometric channels characterized by spatial frequency, phase and orientation.

Recently, one of the authors [Weiman, 18] of the proposal at hand designed a minimal discrete approximation to the Gabor function and discovered several new properties useful for binocular vision. Principal among these properties are 1) Coefficients are small whole numbers which reduces computational complexity, and 2) Phase measurement is accurate to sub-pixel magnitudes and may be computed as a linear function of filter output. Thus, the discrete Gabor filter is efficient yet high on performance.

Figure 1-1 gives a cross-section of the real and imaginary components of the continuous Gabor function, expressed in equation 1.

$$g(x;\ \mu,\sigma,\omega) = Gauss(x;\ \mu,\sigma)\, e^{i\,\omega\, x} \tag{1-1}$$

where

$$Gauss(\,x\,;\,\mu,\,\sigma) = \frac{1}{\sigma\sqrt{2\,\pi}}\exp\left(\frac{(x-\mu\,)^2}{2\,\sigma^2}\right) \tag{1-2}$$

Figure 1-2 illustrates a cross section the new discrete approximation with small integer coefficients. The elevations are magnitudes of coefficients of a filter to be applied to an 8-pixel wide window of imagery. The pair of filters g_c (even symmetry) and g_s (odd symmetry) are to be applied simultaneously to the same neighborhood. Gabor phase, in analogy with the term in Fourier transforms, refers to the relative magnitude of the outputs of the odd and even (imaginary and real) filter outputs. Figure 1-1d plots the trajectory of relative magnitudes of the filters, G_c (x-axis) and G_s (y-axis) as a shadow moves across the domain of the filters. The angle which a particular ray from the origin in this plot makes with the x-axis is known as the phase. Figure 1-3 illustrates the phase trajectory of the discrete Gabor filter pair of figure 1-2 under the same stimulus. The numbered segments in figure 1-3 correspond to the numbered pixels of figure 1-2. That is, progression along the phase trajectory is linear with position in the image plane, and each pixel transit is characterized by a change in the signs of the filter outputs. This is of utmost importance to binocular stereo. Phase corresponds exactly to disparity, which can be measured to a fraction of a pixel, as we shall show in experiments discussed in section 4.1.

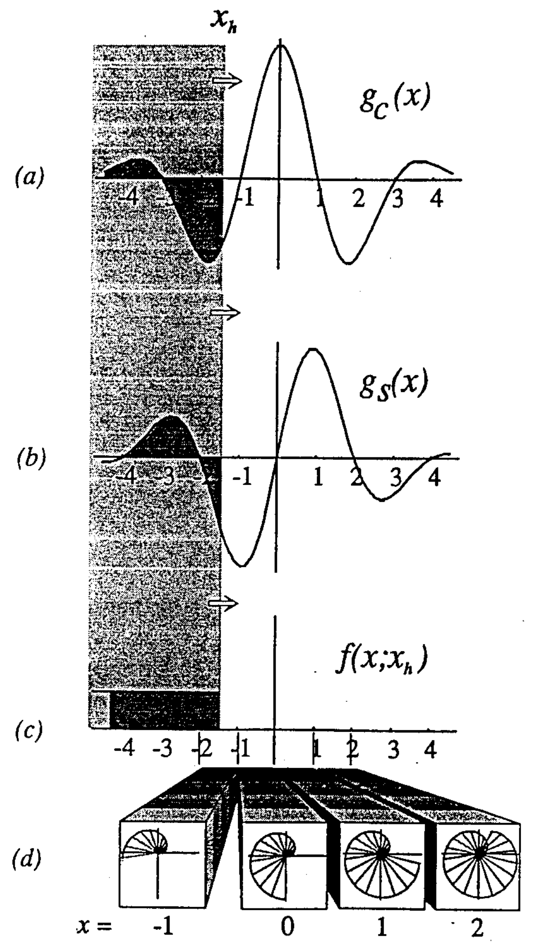


Figure 1-1. Continuous Gabor Function

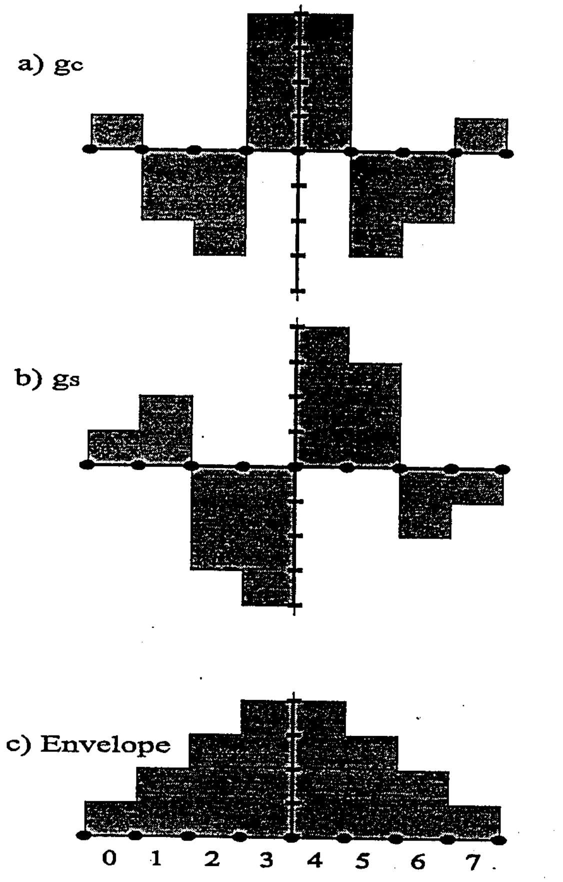


Figure 1-2. Discrete Gabor Function

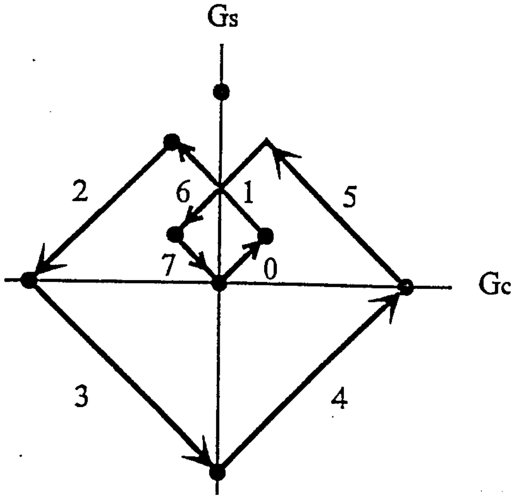


Figure 1-3. Phase Trajectory of Discrete Gabor Filter

And finally, figure 1-4 illustrates the complete 2-D profile of the elementary discrete Gabor function. The steps represent integer coefficients of values +/- 1, +/- 2, +/- 3, +/- 4, and their multiples by factors 2, 3 and 4. Thus a total of four magnitude bits and one sign bit are sufficient to represent the filters.

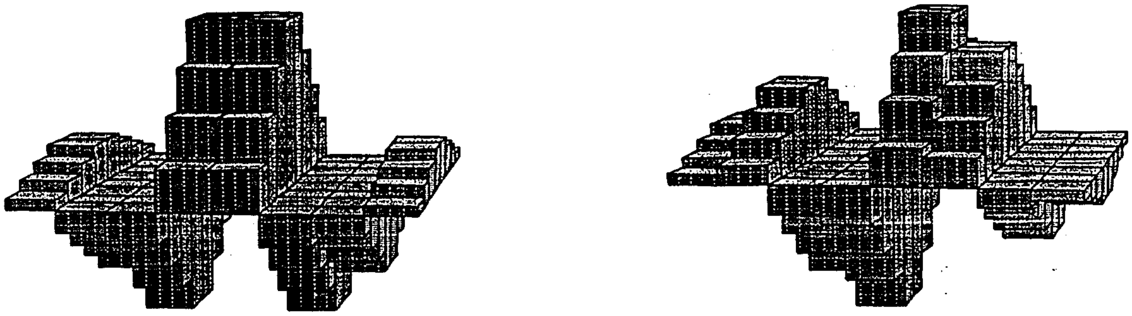


Figure 1-4. 2-D Profile of Discrete Gabor Filter (Even and Odd Components)

These 8x8 filters are to be applied simultaneously to 8x8 windows of the image. Clearly, this pair of filters is maximally sensitive to visual edges parallel to its ridge lines, and has zero response to edges which are perpendicular to the ridges because coefficients cancel symmetrically. Thus, the filters are orientation sensitive. Therefore, to cover all orientations, filters aligned at 0°, 90°, 45° and 135° are necessary. In section 2 we examine the mathematical properties of these filters relevant to image processing tasks for which they were designed. In section 3 we compare their computational cost against alternatives, and in section 4 we describe their application to real imagery in disparity and pattern recognition experiments. Conclusions are summarized in section 5 and references are listed in section 6.

2. SIGNAL PROCESSING AND WAVELET PROPERTIES

In [18] we showed that the discrete Gabor function is "well-behaved" in terms of its ability to faithfully measure energy, frequency, phase. Thus, it is a pleasant surprise that the filter's compact size and severe quantization of coefficients, which give it tremendous computational efficiency, do not introduce distortion, noise, and artifacts which might diminish its performance as a signal processing atom. In this section, we try to extend the examination by asking the question, can the discrete Gabor be used as a universal basis for image decomposition and reconstruction? In more detail, is the set a complete basis, and are the elements orthogonal? Can a wavelet family be constructed?

Positive answers to these questions would provide a clear path to the design of a silicon cortex. In our analysis we found that the answer was negative. The Gabor is not a wavelet. Nevertheless, it is a powerful tool for image analysis and application.

Section 2.1 presents the analysis for continuous mathematics, which is the parent of the discrete case. Several mathematical questions regarding the wavelet properties of the Gabor function are clearly answered in the negative. *(Figure numbers in section 2.1 are local to this section).*

Section 2.2 examines the orthogonality of the discrete Gabor function, and discusses the consequences of non-zero inner products in testing orthogonality. *(Figure numbers in section 2.2 are local to this section).*

Section 2.3 demonstrates the well-behaved bandpass of the discrete Gabor function. *(Figure numbers in section 2.3 are local to this section).*

2.1 Continuous Gabor Function Properties

In recent years, Gabor filter based methods have been applied by several researchers to a variety of problems in machine vision [3, 6, 7, 14, 15]. Gabor functions have optimal location in the joint spatial and frequency domain [6]. They can achieve a lower bound on joint entropy. In addition, psychological and physiological evidence shows that the majority of receptive field profiles of the mammalian visual system match the response of Gabor functions.

Although the Gabor transform has been well recognized as a useful operator, it has not been broadly applied to several applications due to difficulties of computing discrete Gabor coefficients. Gabor functions are not orthogonal, therefore coefficients cannot be found by calculating inner products slone. Bastiaans [1, 2] introduced an auxiliary biorthogonal function and computed Gabor coefficients by projection.

Let $g(x)$ be a normalized function centered at the origin

$$\int \mid g(x) \mid^2 dx = 1. \tag{1}$$

An elementary function of order (m, n) is defined by

$$f_{mn}(x) = g(x - mD) \cdot exp(i \cdot nWx), \tag{2}$$

where m, n are integers, $i = \sqrt{-1}$, and $W \cdot D = 2\pi$. Thus, a signal $\phi(x)$ can be expressed by these elementary functions as

$$\phi(x) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_{mn} \cdot f_{mn}(x). \tag{3}$$

Since the elementary functions are not orthogonal, the analytic formula for calculating coefficients employs an auxiliary function $\gamma(x)$

$$a_{mn} = \int \phi(x) \cdot \gamma^*(x - mD) \cdot exp(-i \cdot nWx)dx, \tag{4}$$

where * denotes complex conjugation. The auxiliary functions that Bastiaans found were

$$\gamma(x) = \Big(\frac{1}{\sqrt{2}D}\Big)^{\frac{1}{2}} \cdot \Big(\frac{K_0}{\pi}\Big)^{-\frac{3}{2}} \cdot exp\Big[\pi\Big(\frac{x}{D}\Big)^2\Big] \cdot \sum_{n+\frac{1}{2}\geq\frac{x}{D}} (-1)^n \cdot exp\Big[-\pi\Big(n+\frac{1}{2}\Big)^2\Big], \tag{5}$$

where K_0 is a normalization factor. Figure 1 shows the auxiliary functions. The major flaw of Bastiaans' approach is that the auxiliary functions may not be well localized, therefore, the Gabor coefficients do not reflect a signal's local behavior.

Qian and Chen [16] developed an orthogonal-like discrete Gabor transform by restricting the auxiliary function to the one that is most similar to a basis. Using this approach, we can obtain coefficients that reflect a signal's local behavior. However, we need to oversample the basis functions, making computational cost high.

Gabor filters have been successfully used in several applications. For example, Friedlander and Porat developed a detection scheme for transient signals of unknown shapes and arrival times, using a Gabor representation. Because the detector was localized, it could detect multiple signals separately. For example, Figure 2(a) shows an original signal. The Gabor coefficients of the signal are shown in Figure 2(b). Figure 2(c) shows the absolute values of the reconstructed signal. (Note : Because the reconstructed signal is complex, we only show its absolute values.)

It is well known that the receptive profiles of simple cells in the mammalian visual system respond close to Gabor filters. Mehrotra *et al.* [12] used Gabor odd filters for step edge detection. Figures 3(a) and (b) show one dimensional Gabor odd and even filters, respectively. Performance analysis revealed that the overall performance of the Gabor filter-based edge detector is almost identical to that of the first derivative of a Gaussian.

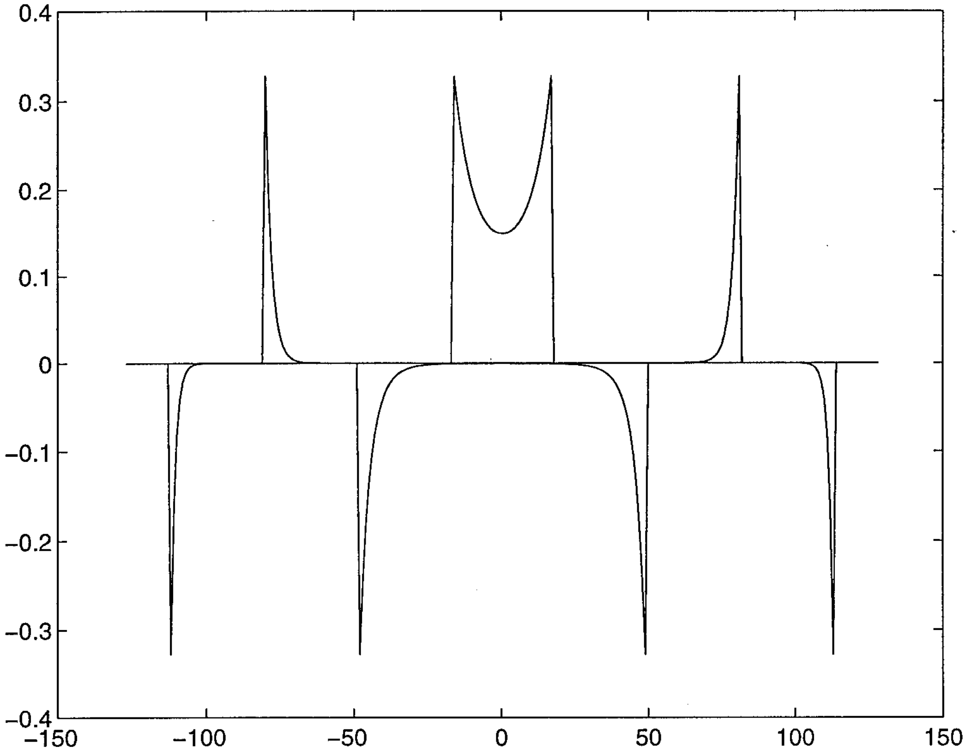
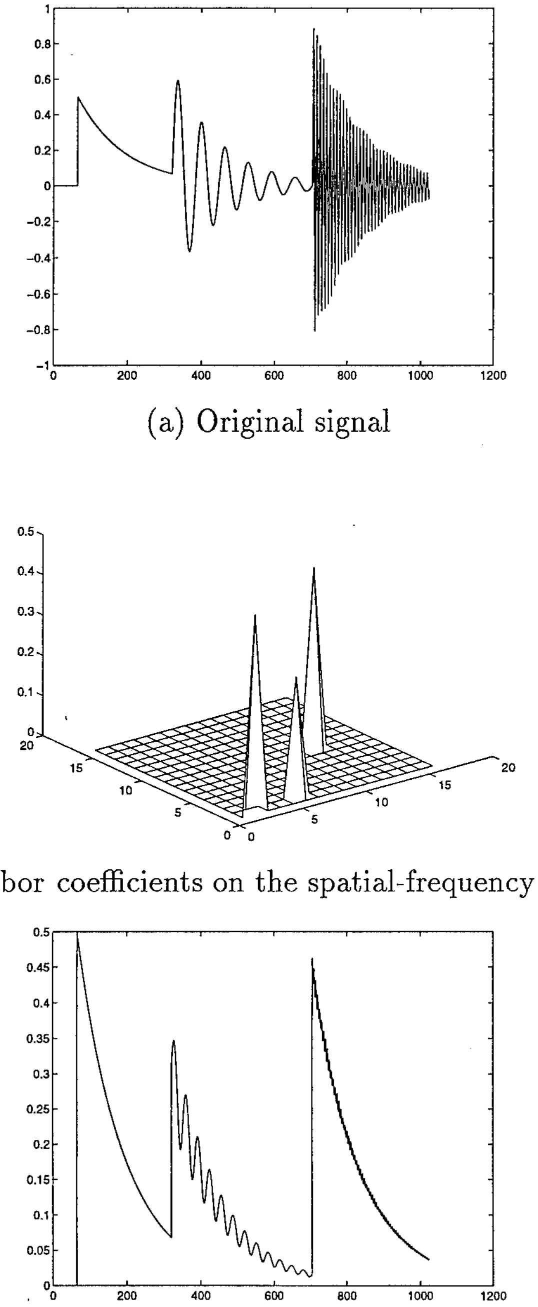


Figure 1: One-dimensional auxiliary functions $\gamma(x)$.

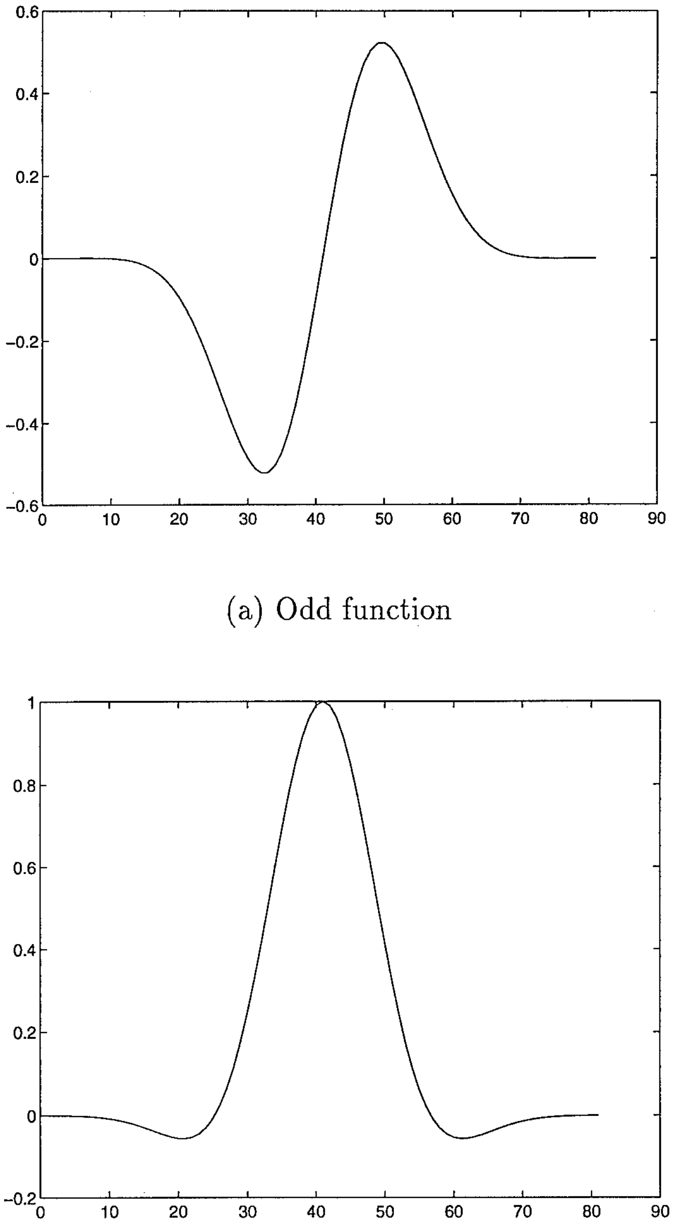


(a) Original signal

(b) Gabor coefficients on the spatial-frequency plane.

(c) Reconstructed signal

Figure 2: Detection of transient signals.



(a) Odd function

(b) Even function

Figure 3: One-dimensional Gabor odd and even functions.

Added to Question 1:

1. *Can an orthonormal basis for $L^2(Z^2)$ be constructed from a multi-resolution composition of 2-D Discrete Gabor Functions (DGF's)?*

ANS:

For clarity, let's consider one dimensional case. For a signal $f(t)$, the Gabor expansion is defined as

$$f(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C_{m,n} h_{m,n}(t), \tag{1}$$

where

$$h_{m,n}(t) = 2^{-\frac{n}{2}} h(2^{-n}t - mT) e^{j\omega 2^{-n}t}, \tag{2}$$

and $C_{m,n}$ are the Gabor coefficients. According to the Balian-Low theorem, $h_{m,n}(t)$ does not form an orthogonal basis unless the corresponding elementary function $h(t)$ is poorly localized in either time or frequency, *i.e.*,

$$\int t^2 h^2(t) dt \int \omega^2 \hat{h}^2(\omega) d\omega \sim \infty. \tag{3}$$

However, a Gaussian function is well localized in time and frequency, and thus cannot form an orthogonal basis. This argument can be extended simply to the two dimensional case.

Added to Question 3:

3. *Is the DGF a wavelet? If yes, what is the wavelet, the scaling function, and the hi-pass signal? If no, is it a quadrature mirror filter? How good or bad are its signal processing characteristics compared to a QMF?*

ANS:

In the one dimensional continuous case, if ψ is a wavelet, it must satisfy the admissibility condition [5]

$$C_\psi = 2\pi \int \frac{|\hat{\psi}(\xi)|^2}{|\xi|} d\xi < \infty, \tag{4}$$

where $\xi \in L^1(R)$. Therefore a function f can be recovered from its wavelet transform via the **resolution of the identity** [5]

$$f = C_\psi^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{dadb}{a^2} (T^{WAV} f)(a,b) \psi^{a,b}, \tag{5}$$

where

$$(T^{WAV} f)(a,b) = |a|^{-\frac{1}{2}} \int dt f(t) \psi(\frac{t-b}{a}). \tag{6}$$

In the discrete case there is no direct analog of the resolution of the identy, and so it is convenient to work with the concept of a **frame**.

A Gabor pair is a *quadrature pair*, but it is not a QMF. Let H_0 and H_1 be the frequency responses of two filters. QMF's are defined to be frequency shifted versions of one another, *i.e.*,

$$H_1(e^{j\omega}) = H_0(-e^{j\omega}) \tag{7}$$

and are also constrained to have even length. For a two-band system, "mirror" part means that their frequency responses are mirror reflected, *i.e.*, they form a highpass filter and a lowpass filter, respectively. But, a Gabor pair (cosine and sine components) have the same frequency responses, and thus can be called only quadrature pairs.

2.2 Orthogonality Properties of the Discrete Gabor Filter

Orthogonal decompositions have proved useful for signal analysis since the invention of Fourier series. Complex data is decomposed into the sum of well-behaved known functions which can be routinely processed. Image analysis, compression, and reconstruction are often based on orthogonal decomposition.

gs0: Filter coefficients of sin 0

1	2	3	4	4	3	2	1
2	4	6	8	8	6	4	2
-3	-6	-9	-12	-12	-9	-6	-3
-4	-8	-12	-16	-16	-12	-8	-4
4	8	12	16	16	12	8	4
3	6	9	12	12	9	6	3
-2	-4	-6	-8	-8	-6	-4	-2
-1	-2	-3	-4	-4	-3	-2	-1

gc0: Filter coefficients of cos 0

1	2	3	4	4	3	2	1
-2	-4	-6	-8	-8	-6	-4	-2
-3	-6	-9	-12	-12	-9	-6	-3
4	8	12	16	16	12	8	4
4	8	12	16	16	12	8	4
-3	-6	-9	-12	-12	-9	-6	-3
-2	-4	-6	-8	-8	-6	-4	-2
1	2	3	4	4	3	2	1

gs90: Filter coefficients of sin 90

1	2	-3	-4	4	3	-2	-1
2	4	-6	-8	8	6	-4	-2
3	6	-9	-12	12	9	-6	-3
4	8	-12	-16	16	12	-8	-4
4	8	-12	-16	16	12	-8	-4
3	6	-9	-12	12	9	-6	-3
2	4	-6	-8	8	6	-4	-2
1	2	-3	-4	4	3	-2	-1

gc90: Filter coefficients of cos 90

1	-2	-3	4	4	-3	-2	1
2	-4	-6	8	8	-6	-4	2
3	-6	-9	12	12	-9	-6	3
4	-8	-12	16	16	-12	-8	4
4	-8	-12	16	16	-12	-8	4
3	-6	-9	12	12	-9	-6	3
2	-4	-6	8	8	-6	-4	2
1	-2	-3	4	4	-3	-2	1

Table 1. Discrete Gabor Filters for 0° and 90°

The discrete Gabor filters illustrated in Tables 1 and 2 *(Note: table and figure numbers following are local to this section)* were proposed as a set of orthogonal functions for image decomposition in a bandpass of one octave about a center frequency of 1/4 cycle per pixel (as will be shown in section 2.3). Orthogonality means that the inner (dot) product of different members of the set is zero; i.e. there is no crosstalk between different members, so that reconstruction of the image can be executed simply by adding these component elements together, multiplied by the coefficients derived from original application to the image.

The filters in Table 1 are clearly orthogonal to each other by inspection of symmetry. The horizontal filters at the top are out of synch by a sign change per row, yielding zero for inner product. The same can be said for the columns of the two vertical filters at the bottom. Furthermore, the orthogonal geometric orientation of elements of the top pair against the bottom pair also suggests zero inner product. Table 3, which lists all possible inner products, shows that this is true for all except the cosines at 90°, whose inner product is 64. The consequences of this non-zero value are no practical importance, as can be seen by comparing them to the magnitudes of the diagonal terms, which indicate the energies of the individual filters. These diagonals range from 2684 to 3600, so 64 is between 2% and 2.5% error, i.e. below the fifth significant bit of pixel value, and can be regarded as zero for all practical purposes.

gs45: Filter coefficients of sin 45

1	2	3	0	-4	-3	-2	0
2	4	0	-8	-8	-6	0	2
3	0	-9	-12	-12	0	6	3
0	-8	-12	-16	0	12	8	4
-4	-8	-12	0	16	12	8	0
-3	-6	0	12	12	9	0	-3
-2	0	6	8	8	0	-4	-2
0	2	3	4	0	-3	-2	-1

gc45: Filter coefficients of cos 45

1	0	-3	-4	-4	0	2	1
0	-4	-6	-8	0	6	4	2
-3	-6	-9	0	12	9	6	0
-4	-8	0	16	16	12	0	-4
-4	0	12	16	16	0	-8	-4
0	6	9	12	0	-9	-6	-3
2	4	6	0	-8	-6	-4	0
1	2	0	-4	-4	-3	0	1

gs135: Filter coefficients of sin 135

0	2	3	4	0	-3	-2	-1
-2	0	6	8	8	0	-4	-2
-3	-6	0	12	12	9	0	-3
-4	-8	-12	0	16	12	8	0
0	-8	-12	-16	0	12	8	4
3	0	-9	-12	-12	0	6	3
2	4	0	-8	-8	-6	0	2
1	2	3	0	-4	-3	-2	0

gc135: Filter coefficients of cos 135

1	2	0	-4	-4	-3	0	1
2	4	6	0	-8	-6	-4	0
0	6	9	12	0	-9	-6	-3
-4	0	12	16	16	0	-8	-4
-4	-8	0	16	16	12	0	-4
-3	-6	-9	0	12	9	6	0
0	-4	-6	-8	0	6	4	2
1	0	-3	-4	-4	0	2	1

Table 2. Discrete Gabor Filters for 45° and 135°

Table 2 lists the coefficients of the diagonal discrete Gabor filters. Since they do not align with the coordinate axes, some violence is done to the symmetry of the odd functions, whose tails are asymmetrically pushed into the corners, and the energy of the even functions, whose tails are clipped in the confines of the corners. These problems can be characterized as discretization noise, and are manifested in the uneven and diminished energy values along the main diagonal of table 3. The odd (sin45 or sin135) diagonal's energy is only 75% of its horizontal and vertical counterparts, and the even (cos45 or cos135) diagonal's energy is only slightly higher. These diminutions could easily be overcome in reconstruction, however, by scaling up by the reciprocal, namely 1.33.

More serious are the non-zero values of the inner products of supposedly perpendicular filters cos45 and cos135. The value of 480 is 18% of the filter energy. This could yield serious distortion in representing diagonal edges. Examination of filters indicates large positive hills in the center which cannot be canceled by the diminished negative products at the periphery. Some artificial flattening of the filter coefficients could reduce this problem, but not eliminate it. Reconstruction artifacts would resemble phantom x-shaped ripples around a contrast feature.

Now, the final problem elements are the interactions between diagonals and horizontal/verticals. Their magnitude in table 3 is 25% of the peak filter energy. These are a natural consequence of the relatively oblique geometric orientations of the elements in question, but the lead to artifacts of ambiguity. That is, and equal strength output from two such elements could be caused by an edge which crosses either their acute or obtuse intersection. Such cases could be disambiguated by comparison with complementary orientation pairs, but this would require some higher processing.

	$sin0^\circ$	$cos0^\circ$	$sin45^\circ$	$cos45^\circ$	$sin90^\circ$	$cos90^\circ$	$sin135^\circ$	$cos135^\circ$
$sin0^\circ$	3600	0	800	0	0	0	-800	0
$cos0^\circ$	0	3600	0	900	0	64	0	900
$sin45^\circ$	800	0	2684	0	800	0	0	0
$cos45^\circ$	0	900	0	2716	0	900	0	480
$sin90^\circ$	0	0	800	0	3600	0	800	0
$cos90^\circ$	0	64	0	900	0	3600	0	900
$sin135^\circ$	-800	0	0	0	800	0	2684	0
$cos135^\circ$	0	900	0	480	0	900	0	2716

Table 3. Inner Products of Discrete Gabor Filter Pairs

We now illustrate some images analyzed and reconstructed via discrete Gabor filters and several other traditional techniques. The reader is left to make her own judgments as to quality.

Figure 1 shows the original images used in our experimental study at distinct sizes of scale.

Figure 2 shows discrete Gabor filter outputs obtained from the filters mentioned in Table 1. Figures 2(a) - (d) show sine components in four directions, horizontal, 45 diagonal, vertical, and 35 diagonal, respectively. Corresponding cosine components are shown in Figures 2(e) - (h).

Figure 3 shows five distinct filters and their corresponding outputs. Figure 3(a) shows the lowpass filter. Three bandpass filters are shown in Figures 3(b) - (d). Figure 3(e) shows the highpass filter.

Figure 4 shows incrementally reconstructed images. Figure 4(a) shows the lowpass component, as shown in Figure 3(f), reconstructed from four directional filter pairs. We then reconstructed a bandpass filtered component, as shown in Figure 3(g), in the same way. This component was then added to Figure 4(a) to obtain Figure 4(b). This procedure was carried out recursively to obtain Figures 4(c) and (d). Finally the highpass component, shown in Figure 3(j), was added to obtain Figure 4(e).

Figure 5 shows Laplacian Gaussian pyramid images using Burt and Adelson's [4] approach. We decomposed the image into 3 levels of scale. Figures 5(a) - (c) show each distinct Gaussian pyramid image. Laplacian pyramid images are shown in Figures 5(d) - (f).

Figure 6 shows incrementally reconstructed images. Figure 6(d) shows the upsampled image of the coarsest level of the Gaussian pyramid. We upsampled the last level of the Laplacian pyramid, Figure 5(f), and added this to Figure 6(d) to obtain Figure 6(c). In the same way, we obtained Figures 6(b) and (a) using Figures 5(e) and (d) in the Laplacian pyramid, respectively. Figure 6(a) shows the final reconstructed image.

Figure 7 shows perfect reconstruction by using the complete Gabor transform proposed by Yao [19]. Figures 7(a) and (b) show one-dimensional signals; Figures 7(c) and (d) show two-dimensional examples of analysis.

Figures 8 and 9 show sine components and cosine components of four directional discrete Gabor filtered image pyramids, respectively. Three levels of analysis are shown in this example.

Figures 10(a) and (b) show highpass and lowpass filtered images of the example shown in Figures 8 and 9, respectively.

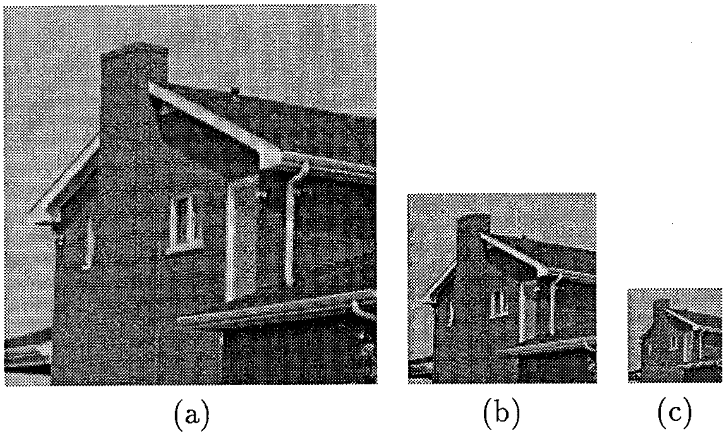


Figure 1: Original images. (a) 256 × 256 (b) 128 × 128 (c) 64 × 64

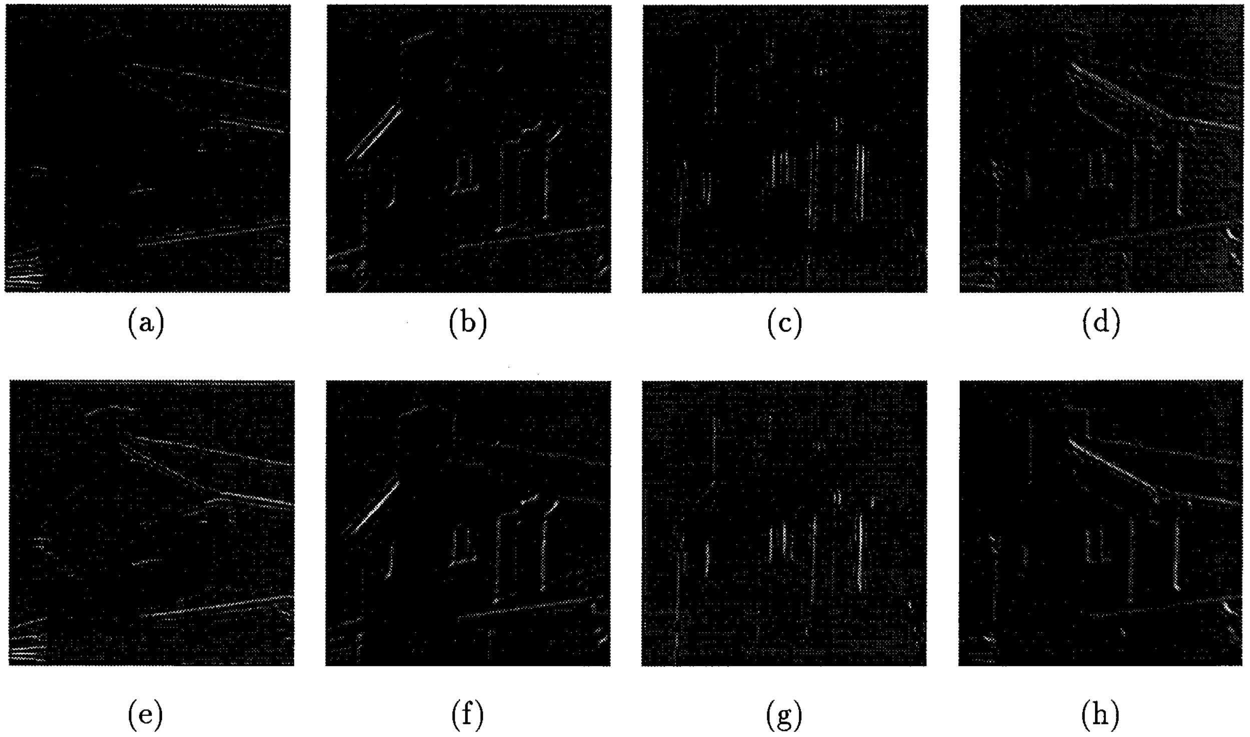


Figure 2: Four directional discrete Gabor filter pairs (→, ↗, ↑, ↖). (a) – (d), *sin* components. (e) – (h), *cos* components.

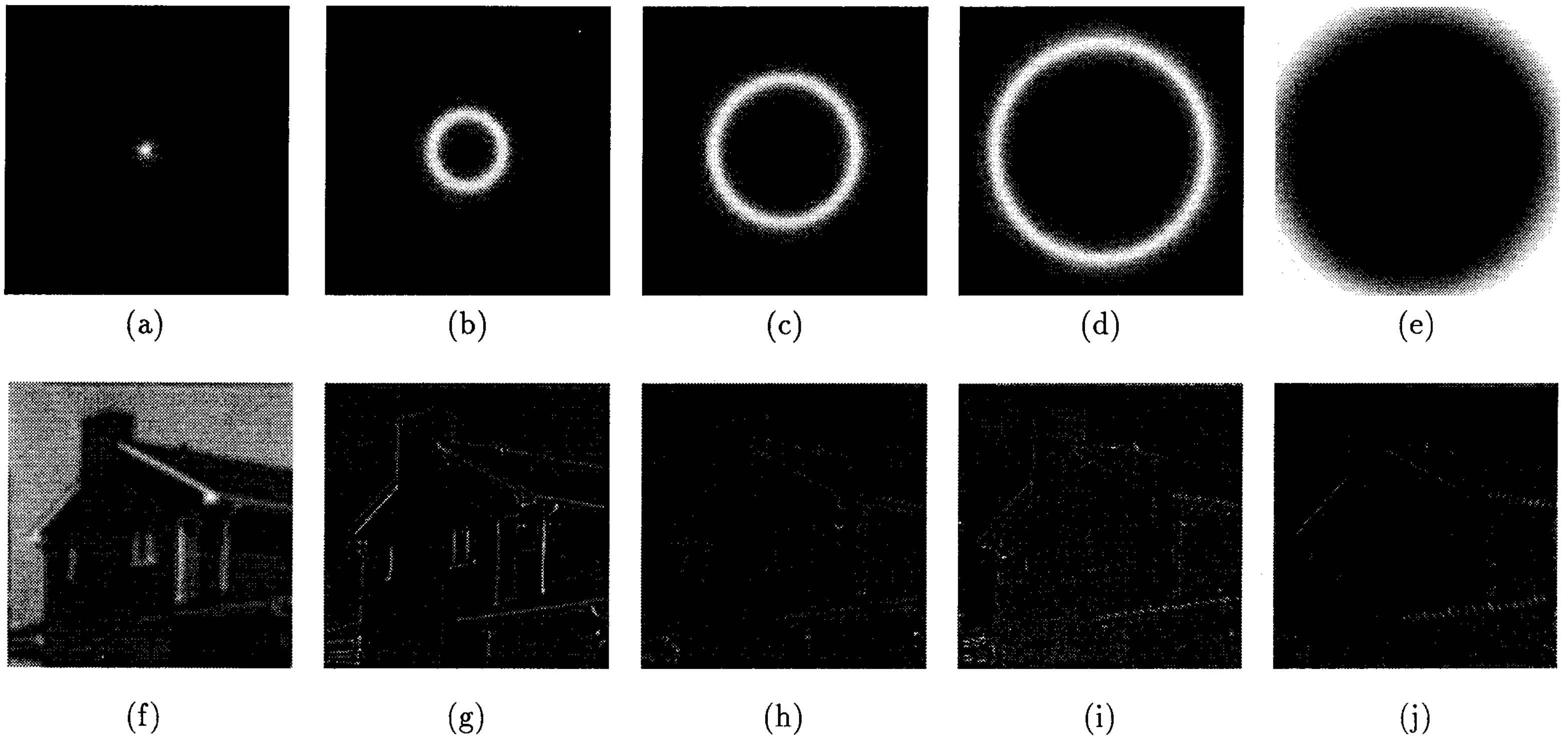
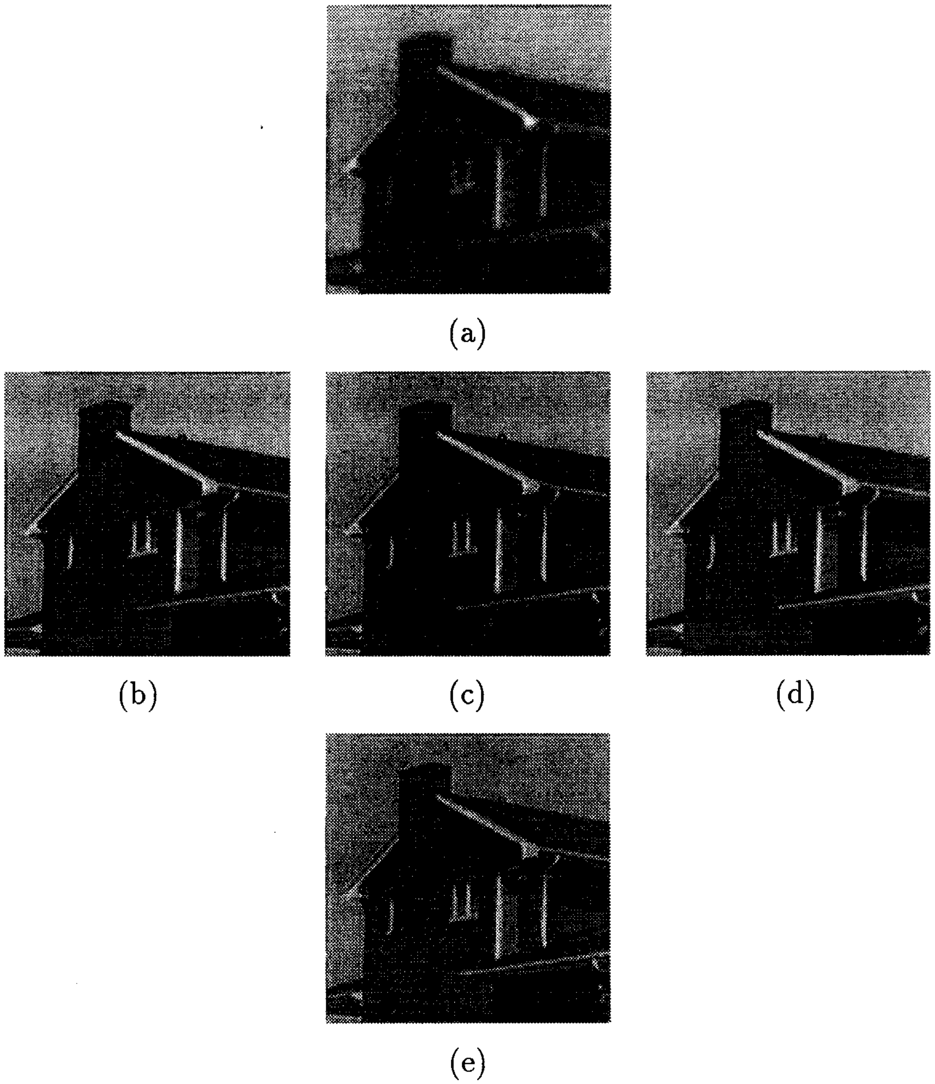


Figure 3: Filter banks and images. (a) – (e), lowpass filter, three passbands, and highpass filter. (f) – (j), corresponding filtered images.



(a)

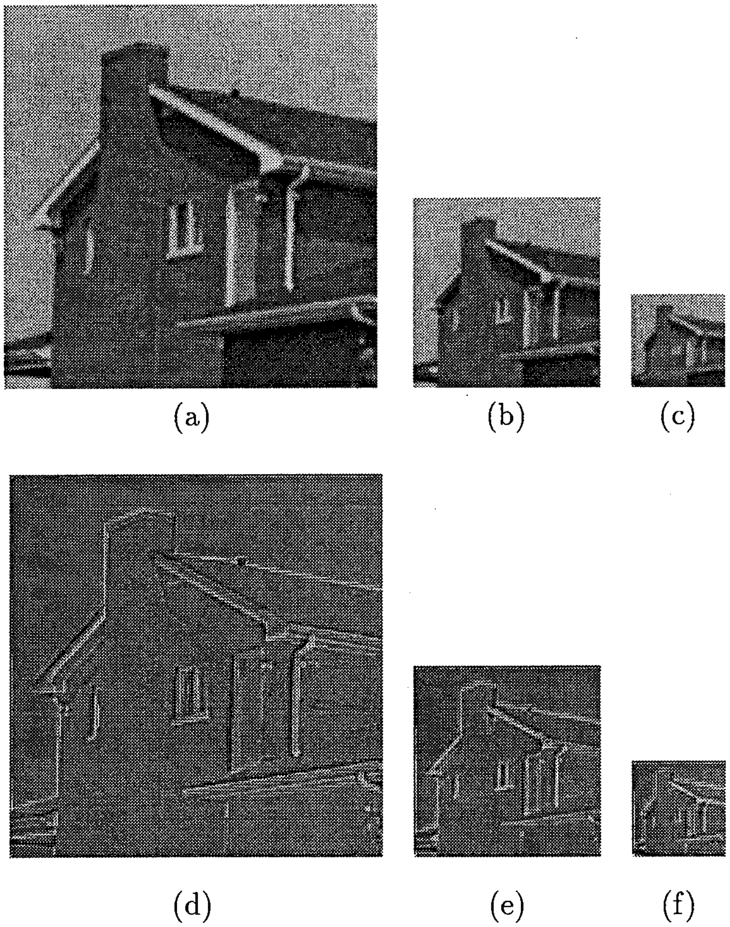
(b)

(c)

(d)

(e)

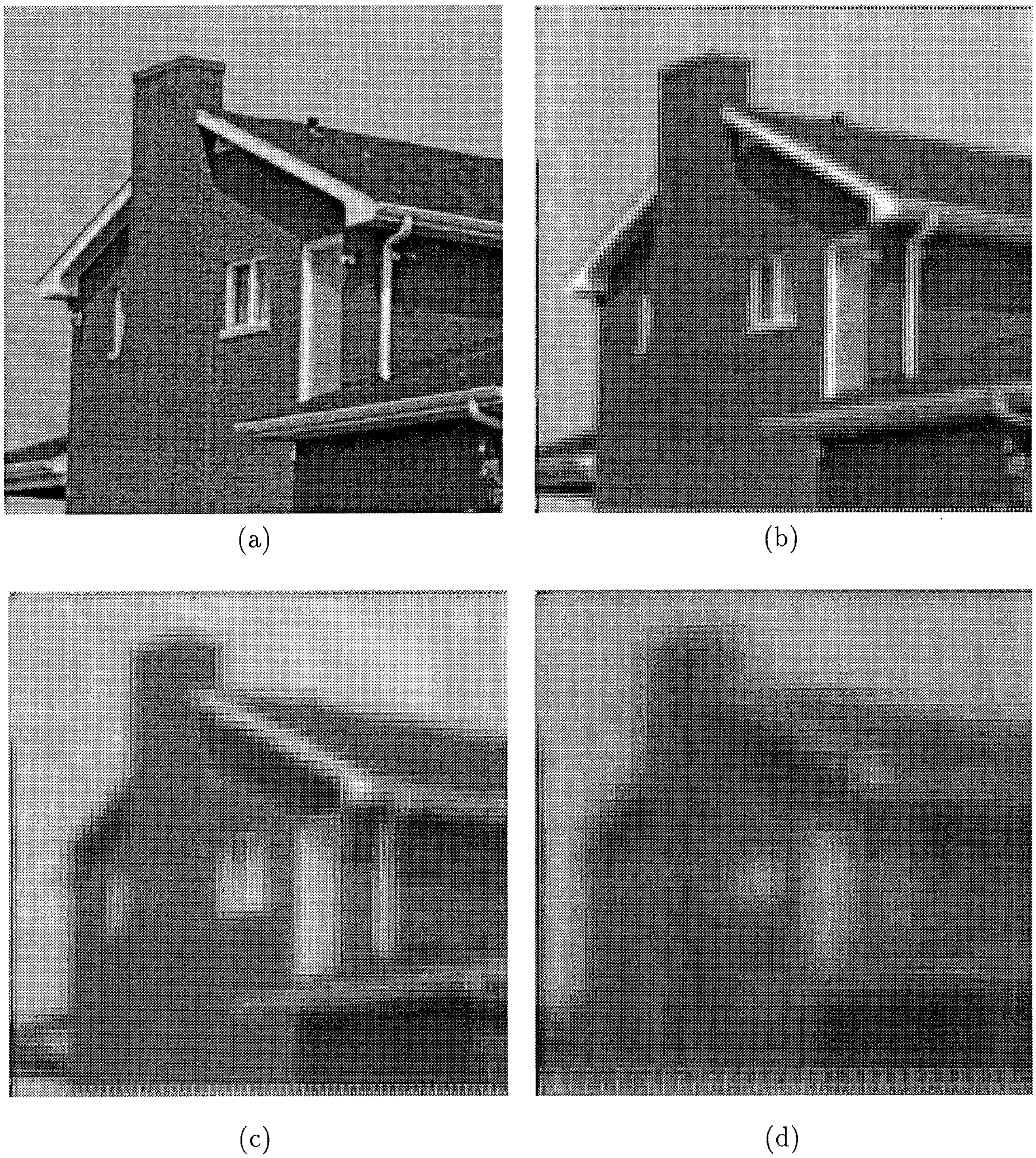
Figure 4: Incrementally reconstructed images.



(a) (b) (c)

(d) (e) (f)

Figure 5: Laplacian Gaussian pyramid images. (a) – (c), Gaussian pyramid images. (d) – (f), Laplacian pyramid images [4].



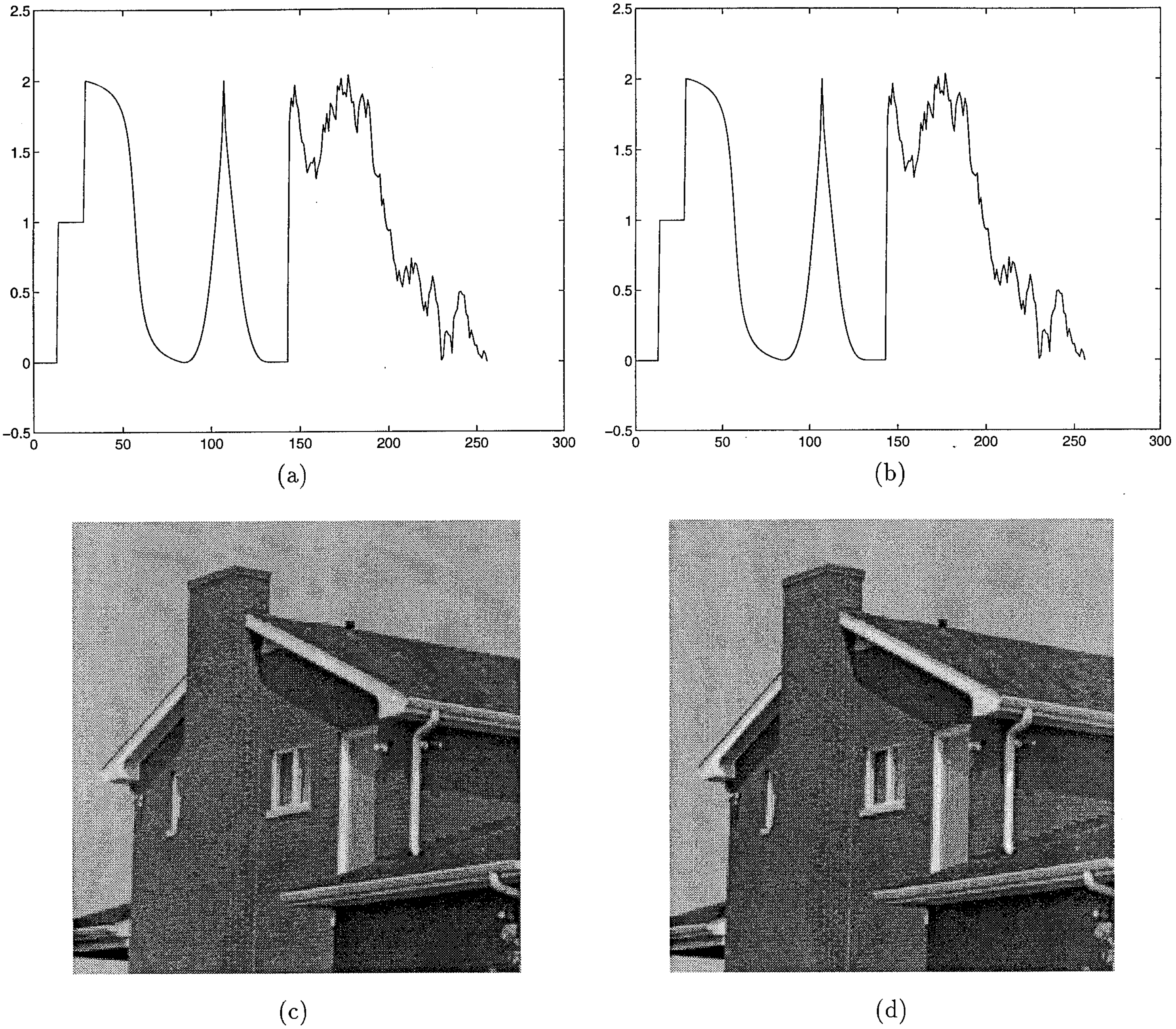
(a)

(b)

(c)

(d)

Figure 6: (a) Reconstructed image. (b) – (d), First three levels of incrementally reconstructed images, via Laplacian pyramid [4].



(a)

(b)

(c)

(d)

Figure 7: Perfect reconstruction by using a complete Gabor transform (single scale) [19]. (a) Original signal. (b) Reconstructed signal. (c) Original image. (d) Reconstructed image.

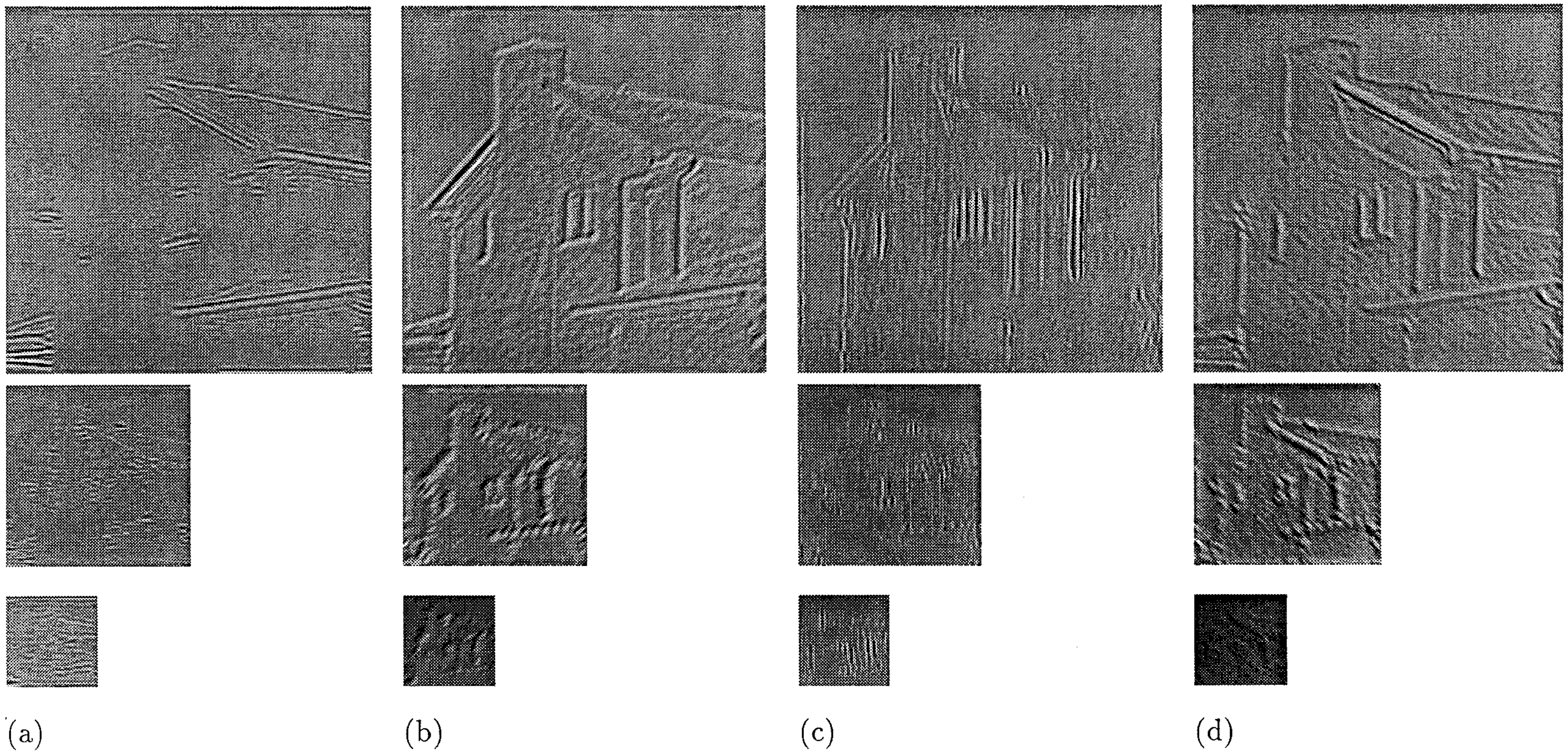


Figure 8: Sine components of a four directional discrete Gabor filtered image pyramid. (a) →, (b) ↗, (c) ↑, (d) ↖, three levels of analysis shown.

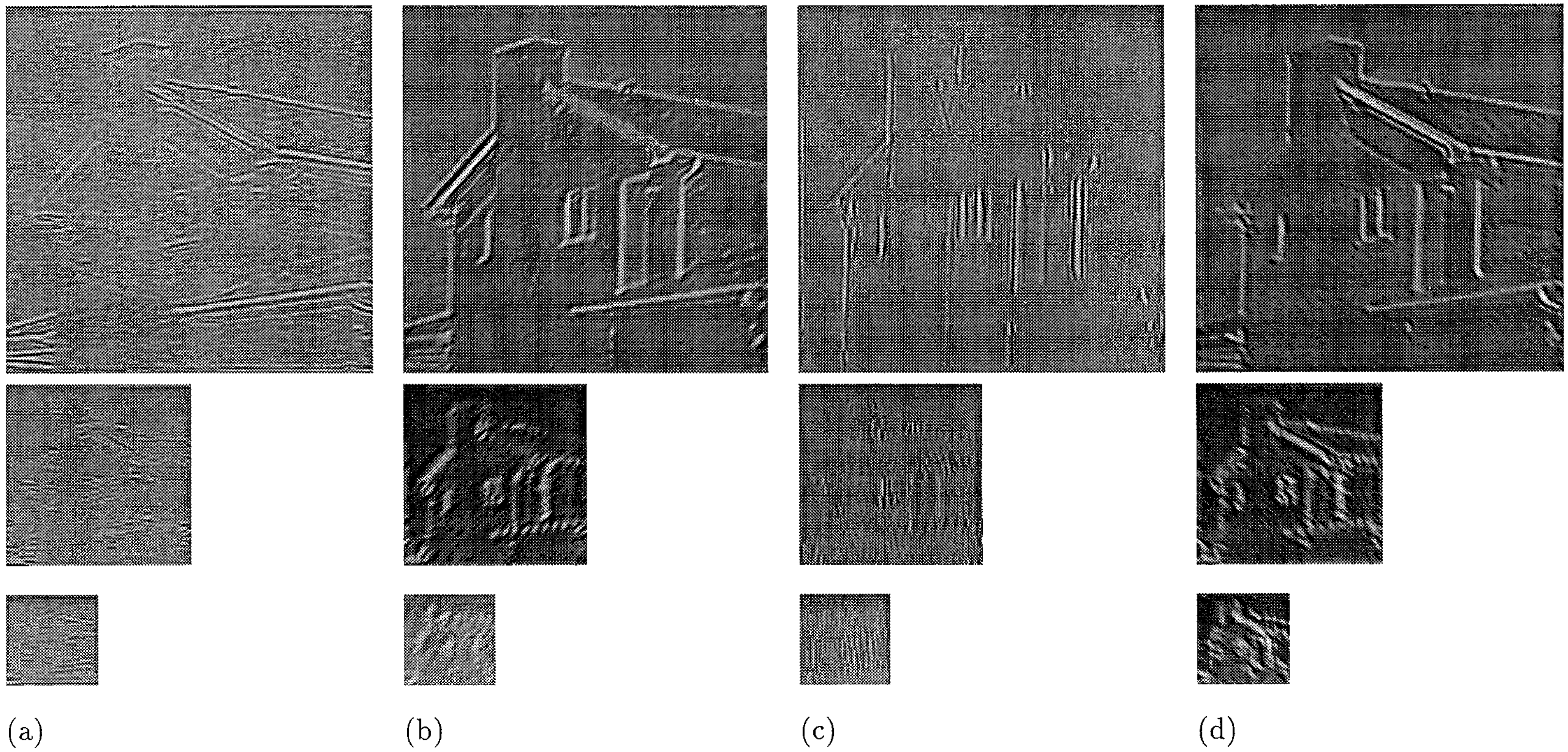
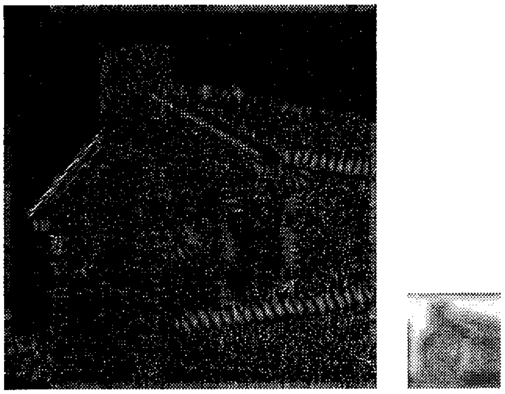


Figure 9: Cosine components of a four directional discrete Gabor filtered image pyramid (a) →, (b) ↗, (c) ↑, (d) ↖, three levels of analysis shown.



(a) (b)

Figure 10: (a) Highpass filtered image. (b) Lowpass filtered image.

2.3 Bandpass of Discrete Gabor Filter

In preceding sections we expressed that the set of discrete Gabor filters responds to sub-pixel phase of arbitrary magnitude and orientation. To complete the picture, we show in the current section that it covers the spatial frequency domain as well. Thus, image edge components of arbitrary position, orientation, and scale can be represented. The discrete Gabor filter is thus "complete" for this class of objects which is universally useful for pattern recognition and stereo ranging.

To show that the Gabor filter responds only to signals within its designed bandpass, namely an octave centered at 1/4 cycle per pixel, we generated a fine-grained sine table in *Mathematica* and then took inner products of its samples at with a one-dimensional cross section of the 2-D Gabor filters' odd and even components. These samples spanned a frequency range between 1/16 and 1/2 pixels per cycle with a grain of 1/32 pixels per cycle. The phase ranged from -4 pixels to 4 pixels with a grain of 1/4 pixel. Figures 1 and 2 below represent the odd and even filter outputs respectively throughout this two dimensional range. The vertical axis represents amplitude of filter output. The leftward horizontal axis is phase in units of .25 pixels, and the rearward axis is frequency in units of 1/32 cycles per pixel. Note that the peak amplitudes are out of phase by two cycles per pixel, which represents 90°, as expected by the phase relationship between odd and even filter components.

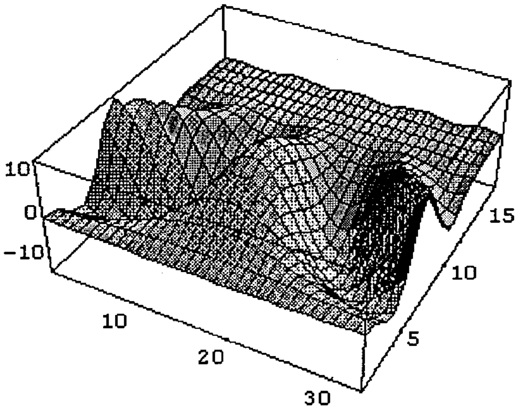


Figure 1. Odd Component Output

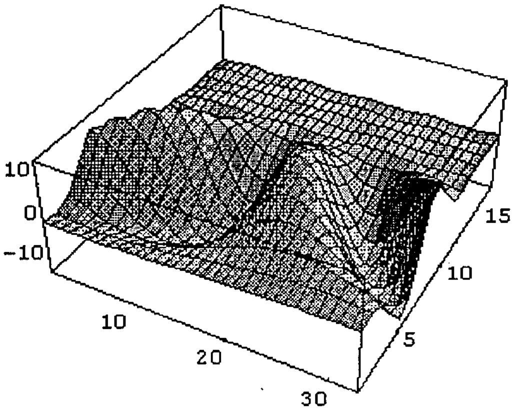
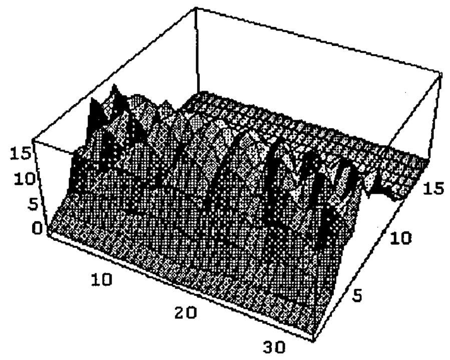


Figure 2. Even Component Output

Adding together the absolute values of these outputs yields figure 3, which shows a high crestline at the center frequency, tapering in nice Gaussian fashion half an octave away in each direction from the central frequency. Note that the deep valleys of the separate phase diagrams are filled in fairly nicely along the entire ridge. Figure 4 illustrates the same data in barchart form from a viewpoint 90° away.



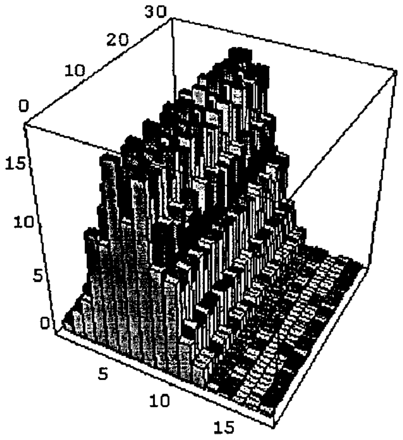
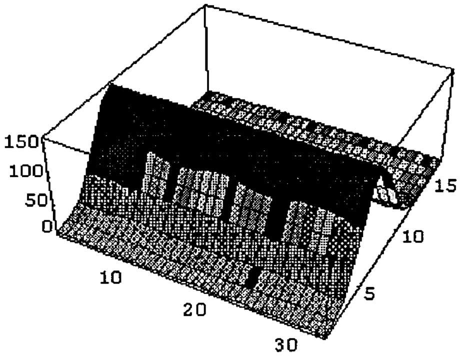


Figure 3. Sum of Abs Energy

Figure 4. Sum of Abs Energy, Barchart

The energy norm for phase calculation in the introduction was specified as sum of absolute values. By using the traditional sum of squares of Gabor filter outputs instead, the figures below appear in place of those above. Note that the energy crest is much smoother; phase ripples are greatly reduced. This may introduce some "bowing" in phase interpretation, i.e. a small fish-eye per pixel measurement error. We do not know what the consequences of this are, but anticipate it would result only in a very small sub-pixel measurement error.



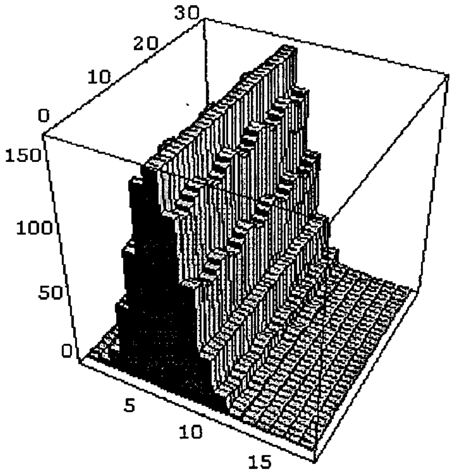


Figure 5. Sum of Squares Energy

Figure 6. Sum of Squares Energy, Barchart

We conclude this section with the observation that the discrete Gabor filter behaves nicely within a spatial frequency bandpass balanced around its center frequency, and that energy is immune to phase variations. Thus, a bandpass pyramided image can be decomposed without aliasing.

3. COMPARATIVE COMPUTATIONAL COMPLEXITY

A conjecture put forth in the proposal was that the discrete Gabor filter would reduce computational complexity compared to alternative but similar image analysis methods. Our findings, described below, indicate that this is not true for the general case, but is true in specific applications which involve orientation tuning. We can summarize as follows. For global transforms such as the Fourier Transform or Karhunen-Loeve, convolutions involving the entire pixel matrix, or eigenvector computations whose dimension is the number of pixels, computational cost far outweighs that of local operators such as LaPlacian of Gaussian and discrete Gabor, so we may dismiss globals from the competition. Local operators are generally windowed filters, whose complexity depends on the size of the window, the density of window application (subsampling) and number of elements which must be applied. The LaPlacian of Gaussians is a single second derivative operator whereas the discrete Gabor involves a set of 8 filters. Thus, all other things being equal (namely, pixel count and window size), the discrete Gabor has a disadvantage of almost an order of magnitude in comparison to the LaPlacian of Gaussian. However, in applications such as stereo ranging, disparity is reduced to a single dimension (along the epipolar lines of the binocular configuration) and the number of filters is reduced to two, compared to the one of the LaPlacian. Because Gabor phase, the critical output in this application, is redundant over the central 4x4 pixels of the window, the density of application can be reduced four-fold in each dimension, subsampling to yield a clear advantage over the operation count of the LaPlacian of Gaussian. Furthermore, subpixel edge location can be derived without interpolation by directly reading the phase of the filter pair.

Having given the preceding qualitative arguments for the advantages of the Gabor in this specific application, we now present quantified arguments for its general disadvantage in computational complexity in the general case, with the caveat that the Gabor does have the opportunity for separating information channels and reducing computation by tuning to the one channel which is relevant, whereas the LaPlacian of Gaussian does not.

The complete Gabor transform is thrown in to the following discussion to show the inefficiency introduced by not restricting the window of application.

Complexity Analysis

In this section, we evaluate the complexity of three algorithms by counting the number of multiplies and additions required for execution. For each method, the decomposition (analysis) algorithm is discussed prior to the reconstruction algorithm (synthesis), and the parameters for image size are denoted by $N \times N$, filter size $M \times M$, and scale L.

I. Laplacian Pyramid

Analysis – We first generated two pyramids: a Gaussian pyramid and a Laplacian pyramid. Each is obtained by convolving a filter with an image. Each pyramid required $2M^2N^2$ multiplications and additions.

Synthesis – The coefficients were first upsampled and convolved with the filter to reconstruct the original image. Thus $2M^2N^2$ multiplications and additions were needed to carry out this operation.

Total Cost – $4M^2N^2$ multiplications and $4M^2N^2$ additions.

II. Complete Gabor Transformation

Analysis – To compute Gabor coefficients, we needed to carry out six matrix multiplications. Each operation required N^3 multiplies and additions, therefore, a total of $6N^3$ operations were needed.

Synthesis – Here, sum up N^2 Gabor basis to reconstruct an original $N \times N$ image. Thus, N^4 multiplications and additions were needed.

Total Cost – $6N^3 + N^4$ multiplications and $6N^3 + N^4$ additions.

III. Discrete Gabor filters

Analysis – First, the image was decomposed by a set of passband filters. This required $L \cdot NlogN$ operations. Each image passband was then convolved with four distinct pairs of Gabor filters. Each of these required $2M^2N^2$ operations.

Synthesis – The four pairs of filtered components and lowpassed component were added together to form a single pyramid. This took $16N^2$ additions. The pyramid can be used to reconstruct an image with $2M^2N^2$ operations.

Total Cost – $L \cdot NlogN + 18M^2N^2$ multiplications and $L \cdot NlogN + 18M^2N^2 + 16N^2$ additions.

The complexity analysis of each algorithm is summarized in Table 3-1.

	Laplacian Pyramid	Complete Gabor	Discrete Gabor filters
Multiplication	$4M^2N^2$	$6N^3 + N^4$	$L \cdot NlogN + 18M^2N^2$
Addition	$4M^2N^2$	$6N^3 + N^4$	$L \cdot NlogN + 18M^2N^2 + 16N^2$

Figure 3-1 Complexity Analysis of Three Algorithms

4. PERFORMANCE ON APPLICATIONS

This section describes experiments performed on real world data in two very different applications. The first was the measurement of disparity for binocular stereo using Gabor phase. the second was the recognition of hand-printed characters using Gabor decomposition of the image as input to a neural net to be trained on the characters.

The disparity experiments were conducted on images viewed through a CCD camera feeding into a frame grabber. A set of procedures was devised for calibrating pixel size and position, and then measuring displacements (disparities) to a fraction of a pixel. The results showed that despite the inherently poor resolution of analog video signals, Gabor phase was able to localize edge position to subpixel precision. This was a very positive result and opens the door to efficient binocular stereo. We did not have the resources to demonstrate the stereo application, but did demonstrate that the essential ingredients are present.

The character recognition application did not show good convergence of neural net recognition using discrete Gabor function inputs, but recognition did occur. This partial result calls for more thorough investigation in later studies. It is possible that the discrete Gabor function is simply too coarse for the task.

The following paragraphs describe experimental procedures and results.

4.1 Gabor Phase for Disparity Measurement

Theoretical analysis and simulation are fruitful in delimiting the potential of the discrete Gabor filter as a framework for image analysis computation. Tests on real imagery are a necessary complement to this analysis, for practical application of the techniques derived. The research proposal specified appliying the Gabor function to the measurement of disparity for binocular stereo vision. To this end, we designed a set of experiments for measuring the positions of edges using Gabor phase, described below.

Experimental Setup:

We set up a Javelin JE7262 CCD camera with 8.5 mm C-mount lens 5 feet (1.52m) away from a target board. Figure 4-1 illustrates the setup. The targets consisted of test patterns generated by a drawing program, printed in gray-scale on white paper on a standard laser printer. The CCD camera video output was input to a Data Translation 1451 frame grabber in a Force SparcStation VME slot. We wrote C programs to grab images, window selected areas for numerical display in matrix form, and compute discrete Gabor phase. The application of these is described below.

The purpose of the experiments was to determine Gabor phase response to a set of patterns expressing a variety of scales and contrasts. Our expectation that Gabor phase would give superb sub-pixel resolution under a variety of illumination conditions was borne out by experiment.

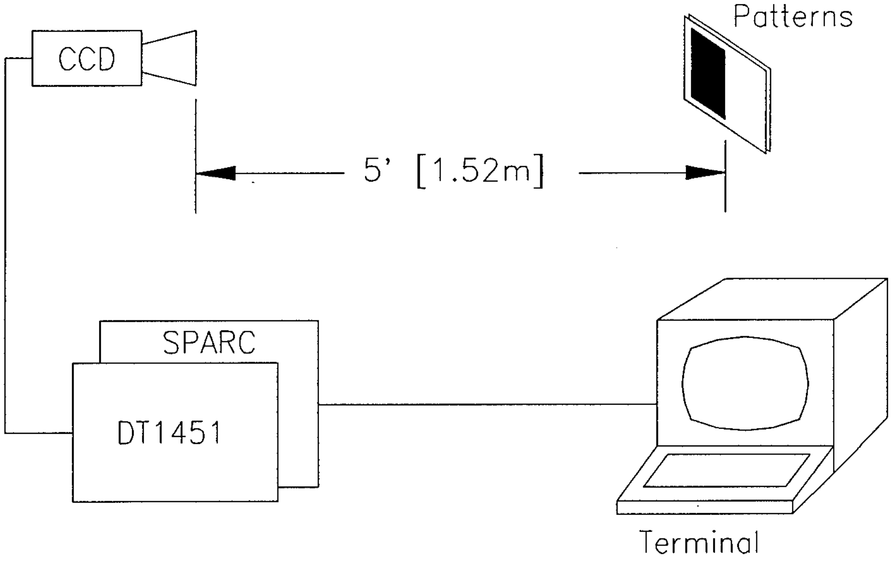


Figure 4-1. Experimental Setup

Experimental Procedure

We designed six patterns illustrating various geometries and contrasts. These were mounted on the target board and images grabbed. A 20x20 window of each image was stored in a file whose name was the sequential number of the pattern in our set, prefixed by the letter "s" for "scene". Most of these were later culled either because they were redundant with others, or irrelevant to the discussion. The survivors were further processed by Gabor phase, to yield 20x20 pixel files whose names were suffixed with the letter "g" for Gabor. A list of the relevant files to be discussed is shown in table 4-1 below. In the discussion, we use the file names in the figure labels for clarity, proceding in numerical order with gaps as noted in table 4-1.

File Name	*Description*
s0	1" strip, .1" disparity
s0g	Gabor phase of same
s3	Shady Checkerboard
s3g	Gabor phase of same

Table 4-1. File Names for Scene Patterns

In addition to the static patterns listed above, we mounted a vertical black/white edge pattern on a sliding stage which could be moved in 1mm increments. Labelling this pattern s7, we moved it from its initial zero reference position in 1mm increments to 5mm, storing each of the 20x20 windowed data in files named s70, s71, s72, s73, s74, s75. Applying Gabor phase to each of these yielded files named s70g, s71g, s72g, s73g, s74g, s75g. Results are described in the following section.

Experimental Results

Scene 0: Calibration of Pixel Position and Size

Scene 0, illustrated in figure 4-2 consists of a 1" wide (25.4mm) vertical black strip with a .1" (2.54mm) "jog" halfway down the right side. This image serves to calibrate pixel size and establish the registration of Gabor phase with pixel position. Figure 4-3 illustrates the brightness values in a 20x20 pixel neighborhood of the jog in the line. The black strip is clearly visible as a contiguous band of columns with depressed pixel values. Note that pixel values in the transition between these two regions are smeared out over a band 5 pixels wide, an artifact of the low-pass transfer function of the analog video input to the frame buffer. The brightness gradient is presented as elevation in the plotted and barcharted images of figures 4-4 and 4-5. Note the wrinkle along the right-hand wall of the valley, corresponding to the .1" jog on the right side of the pattern.

Figure 4-2 (s0)

222	221	220	209	173	127	92	75	66	62	64	85	130	179	208	219	224	225	226	226
223	222	220	209	173	127	90	72	62	60	64	86	130	179	207	218	222	223	224	224
223	222	221	209	174	128	92	73	64	62	67	90	133	180	207	218	224	225	226	225
222	221	220	209	171	124	87	70	61	60	65	90	134	182	208	218	222	223	224	225
224	223	222	211	177	131	94	75	66	63	67	89	132	180	207	218	223	225	226	225
222	221	220	209	173	126	91	75	67	62	64	83	127	176	206	218	223	223	225	225
223	222	221	210	175	128	91	73	66	63	66	88	130	179	207	219	223	225	225	225
223	222	221	209	173	126	89	71	64	61	63	83	128	178	207	219	224	225	226	225
224	223	222	210	172	127	93	76	66	62	64	87	132	180	208	219	223	225	226	226
222	222	220	208	172	124	88	71	64	62	63	79	120	169	203	217	223	224	225	225
223	222	222	210	174	129	96	79	70	64	64	72	102	147	190	211	221	224	226	225
225	223	222	209	172	124	88	72	65	63	64	71	98	143	187	210	220	223	225	224
226	224	223	211	174	127	91	73	64	63	64	70	99	144	188	210	220	223	226	225
225	224	224	212	177	130	94	76	67	62	61	66	92	137	183	207	218	222	224	222
226	225	223	211	173	126	91	74	66	65	66	72	98	141	186	209	219	222	224	224
224	223	222	211	176	128	91	73	63	60	61	67	92	135	182	206	217	221	223	223
225	223	222	211	177	130	93	74	65	62	63	68	94	138	183	208	219	222	225	225
222	221	221	210	176	130	93	74	63	60	61	66	89	130	178	205	217	220	222	223
223	222	221	210	175	129	93	76	67	62	61	64	87	130	179	205	217	221	223	223
222	221	219	209	174	126	88	70	63	60	60	63	87	130	179	205	216	220	223	223
223	222	220	209	173	127	90	72	62	60	64	86	130	179	207	218	222	223	224	224

Figure 4-3 Raw Image Data for 20x20 Pixel Window in s0.

```
0]:=
 ListPlot3D [Table[list[[i]], {i, 20}], ViewPoint->{-0.037,-1.735,2.905} ] ;
```

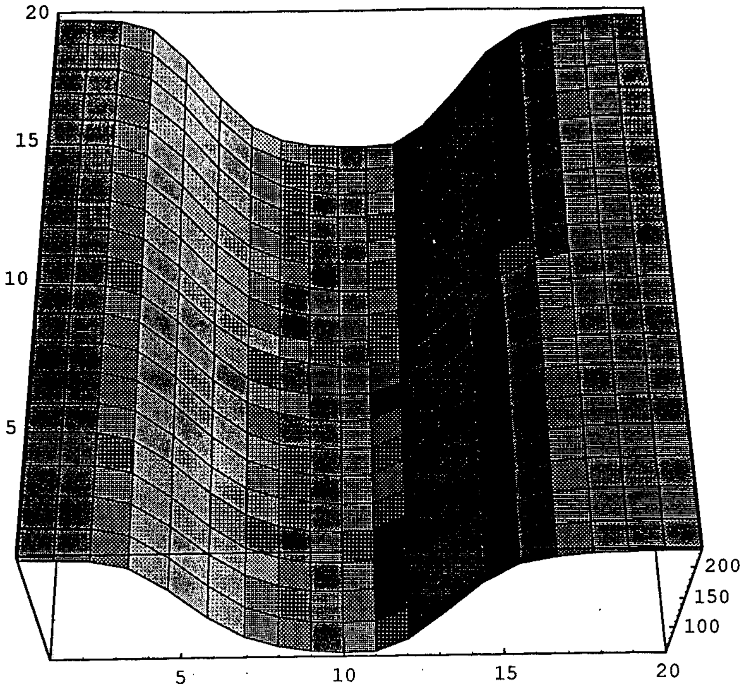


Figure 4-4 Plot of Raw Data in 20x20 Pixel Window in s0.

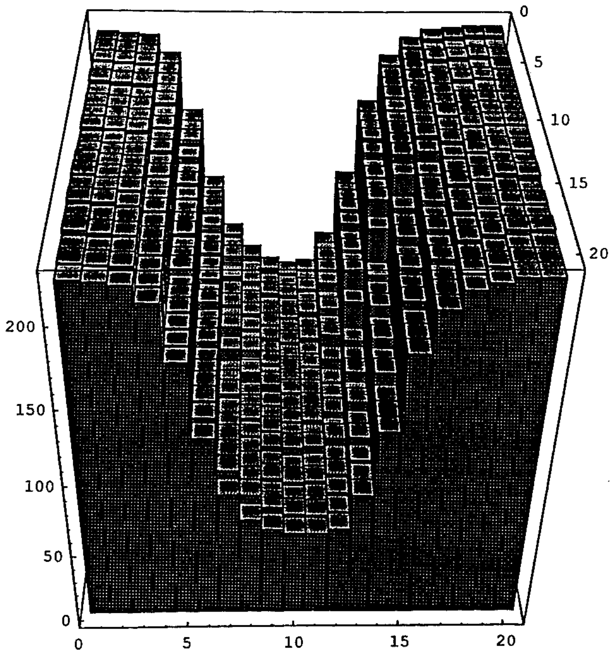


Figure 4-5. Barchart of Raw Data in 20x20 Pixel Window in s0.

We find the precise position of the edge in this smeared data by interpolation as follows. Using 220 as representative of pixel values in the white region and 60 in the black region, the average of these, 140, can be used as pixel value of the boundary, assuming linearity of pixel response. We estimate subpixel position of the actual boundary by interpolating between the pixel values which straddle 140 at the top of the figure, namely columns 5-6 and 13-14. The interpolation equation used is

$$c_i = \frac{140 - v_1}{v_2 - v_1} + c_1 \tag{4-1}$$

where c_i is the pixel index of the boundary (to subpixel accuracy), c_1 is the index of the pixel to the left of the boundary, and v_1 and v_2 are the pixel values which straddle 140, where is v_1 associated with c_1. Substituting the values of pixels in the top row of figure 4-3 into equation 4-1 yields left and right boundaries of 5.72 and 13.2 in subpixel coordinates, for a boundary strip width of 7.48 pixels, which corresponds to .13" (3.4mm) per pixel. This measurement is corroborated by the extended width of the black strip below the jog, measured at row 15, columns 13-14, which interpolates to about .77 pixels, i.e. very close to the .75 predicted by the calibration above. Table 4-2 below summarizes these calibration results.

Numerator Units				
	pixel	*inch*	*.1 inch*	*mm*
per (below)				
pixel	1	.134		3.39
inch	7.5	1		25.4
.1 inch	.75		1	2.54
mm	.29	.039		1

Table 4-2

Having derived the pixel calibration and registration above, we now apply the vertical 8x8 digitized Gabor filter of figure 1-xx to the data in figure 4-3. We reduce the complexity of the depiction by screening out as invalid, all Gabor outputs whose energies are small, and restricting the display to phases between 0 ° and 180 °. The result is shown in figure 4-6, with Gabor outputs displayed at the pixel which is in registration with the upper left corner of the 8x8 window. Since the geometric position of the edge detected by Gabor phase is actually at the center of the 8x8 window, the Gabor array is offset from the raw data array by -3.5 pixels in row and column.

0	0	0	0	0	0	0	120	104	83	67	57	53	0	0	0	0	129	129	47
0	0	0	0	0	0	0	121	104	83	67	57	53	0	0	0	0	134	134	47
0	0	0	0	0	0	0	121	105	84	67	57	53	0	0	0	0	134	134	47
0	0	0	0	0	0	0	122	106	85	68	58	53	0	0	0	0	134	134	47
0	0	0	0	0	0	0	122	107	87	69	59	53	0	0	0	0	134	134	46
0	0	0	0	0	0	0	123	109	89	71	60	54	0	0	0	0	133	134	46
0	0	0	0	0	0	0	125	111	92	73	61	55	0	0	0	0	133	133	45
0	0	0	0	0	0	0	126	113	95	76	63	55	0	0	0	0	132	133	45
0	0	0	0	0	0	0	127	114	98	78	64	56	0	0	0	0	132	132	46
0	0	0	0	0	0	0	128	115	100	79	65	56	0	0	0	0	132	132	47
0	0	0	0	0	0	0	129	116	100	80	65	57	0	0	0	0	131	132	47
0	0	0	0	0	0	0	129	116	101	81	66	57	44	0	0	0	130	132	47
0	0	0	0	0	0	0	130	117	102	81	66	57	44	0	0	0	130	131	47
0	0	0	0	0	0	0	130	117	102	82	66	57	45	0	0	0	125	0	47
0	0	0	0	0	0	0	130	118	102	82	67	57	46	0	0	0	0	0	0
0	0	0	0	0	0	0	131	118	103	82	67	57	48	0	0	0	0	0	0
0	0	0	0	0	0	0	131	118	103	82	67	57	50	50	0	0	0	0	0
0	0	0	0	0	0	0	131	118	103	82	67	57	52	54	0	0	0	0	0
0	0	0	0	0	0	0	131	118	103	82	67	57	53	55	0	0	0	0	0
0	0	0	0	0	0	0	130	118	103	83	67	57	54	55	0	0	0	0	0

Figure 4-6. Gabor Phase for Figure s0

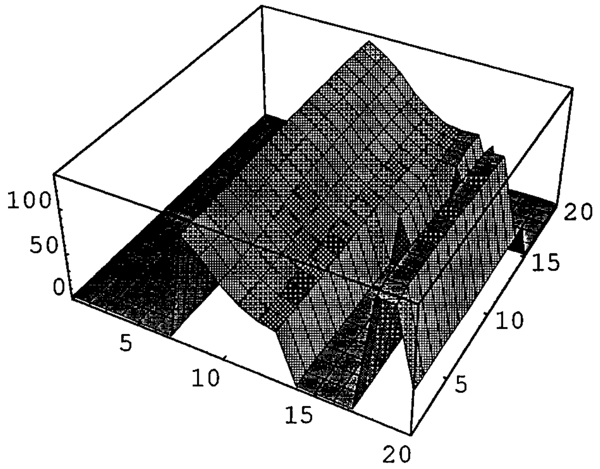
Recall from the description of the discrete Gabor filter that a phase value of 90 degrees corresponds to the maximum filter energy output, triggered by an edge between two pixels at the center of the filter. Theoretically, the values on either side of 90 should increment leftward by 90 degrees for every pixel because the Gabor wavelength is 4 pixels. Instead, they go by 16-21 degrees. This is a consequence of the extremely poor bandpass of the analog video signal, illustrated by the slow rise across the sharp boundary of figure 4-2. The solution for this problem is to use digital cameras rather than analog video for input. We circumvent this problem by localizing the edge to sub-pixel accuracy by interpolating for a phase value of 90°, just as we did before in equation 4-1, which now becomes

$$c_i = \frac{90 - v_1}{v_2 - v_1} + c_1 \qquad (4\text{-}2)$$

Calculating edge position using row 1 of figure 4-6 yields subpixel position 9.61 in that reference frame. Adding 3.5 to register with the raw image data in figure 4-3 yields subpixel position 13.17. This is only 3/100 of a pixel away from the 13.20 calculated by interpolating raw pixel values. Figure 4-7 plots the Gabor phase data of figure 4-6 as elevation, and figure 4-8 as shading.

In[4]:=

```
ListPlot3D [%]
```



Out[4]=

```
-SurfaceGraphics-
```

Figure 4-7. Elevation Display of Gabor Phase (file s0g) of Scene 0.

```
ListDensityPlot[datax]
```

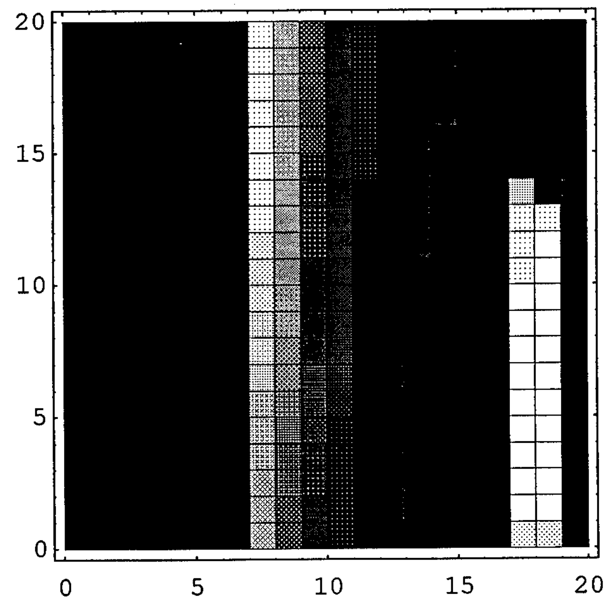


Figure 4-8. Shading Display of Gabor Phase (file s0g) of Scene 0.

Scene 3: Gabor Phase Stability Under Illumination Variations

Scene 3, illustrated in figure 4-9, demonstrates the relative insensitivity of Gabor phase to illumination and contrast variation. The top half of the figure illustrates a low-contrast vertical boundary, the bottom half, a high contrast vertical boundary. Figure 4-10 gives the numerical values and figure 4-11 plots these values as elevation. Note the four different levels of plateaus, separated by sharp slopes in the y-dimension (interline video) and shallow slopes in the x-dimension (interpixel video), emphasized in the viewpoint of figure 4-12. The barchart of figure 4-13 emphasizes the magnitude of the steps more precisely.

Figure 4-9. Checkerboard

60	60	61	62	62	62	63	63	63	65	78	111	153	183	197	201	203	203	205	205
59	58	59	59	59	59	60	61	62	63	75	106	148	180	195	200	203	203	205	204
60	61	63	63	62	63	65	65	64	65	78	111	153	183	197	201	203	203	204	204
58	59	61	61	60	60	60	61	63	65	77	109	150	181	196	201	203	203	204	205
61	61	60	59	60	63	65	66	67	71	84	114	154	183	197	201	204	205	206	206
60	60	60	61	62	65	65	64	62	62	74	107	150	181	196	201	203	203	204	204
61	59	59	60	63	64	65	67	68	68	80	111	152	183	197	201	203	203	205	205
57	57	58	59	61	61	61	60	60	60	73	106	149	181	196	201	203	202	204	204
64	64	63	62	63	65	66	66	65	67	80	111	151	182	197	202	203	203	205	205
60	61	62	63	62	61	61	61	62	65	79	110	150	181	196	200	203	204	206	206
60	60	61	61	62	63	63	65	65	67	80	112	152	182	197	202	204	204	206	207
61	60	61	64	66	66	66	67	68	70	86	122	167	198	213	219	222	222	225	225
51	51	53	55	55	56	56	56	57	62	82	123	173	205	221	226	229	230	231	231
49	49	51	50	50	51	53	55	57	60	81	124	175	207	222	227	229	229	231	230
53	53	53	53	52	52	53	54	56	61	83	127	177	209	224	229	230	230	231	231
50	49	49	48	49	53	57	59	59	59	78	121	174	206	222	227	229	229	230	231
52	52	54	56	56	55	56	59	60	62	81	122	173	206	223	228	231	232	233	232
53	53	54	53	52	54	55	54	54	59	81	126	177	209	224	229	231	231	232	231
53	53	53	54	55	55	55	56	58	62	83	126	176	208	224	229	232	231	232	232
51	50	52	54	56	55	55	56	57	61	82	125	176	208	223	227	229	229	231	231

Figure 4-10. Checkerboard Data of File s3

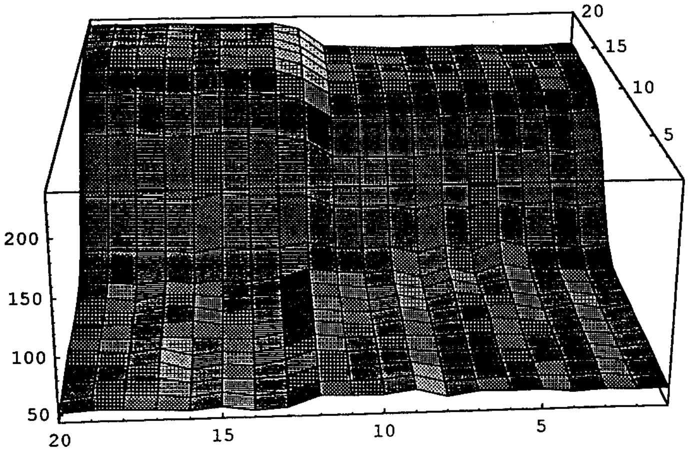


Figure 4-11. Elevation Plot of Checkerboard

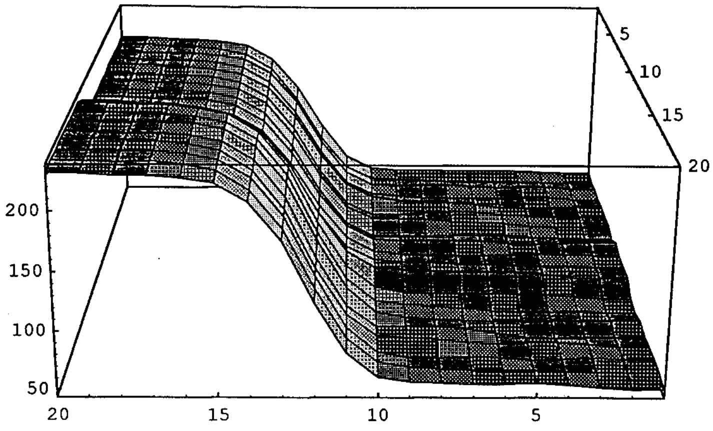


Figure 4-12. Elevation Plot of Checkerboard

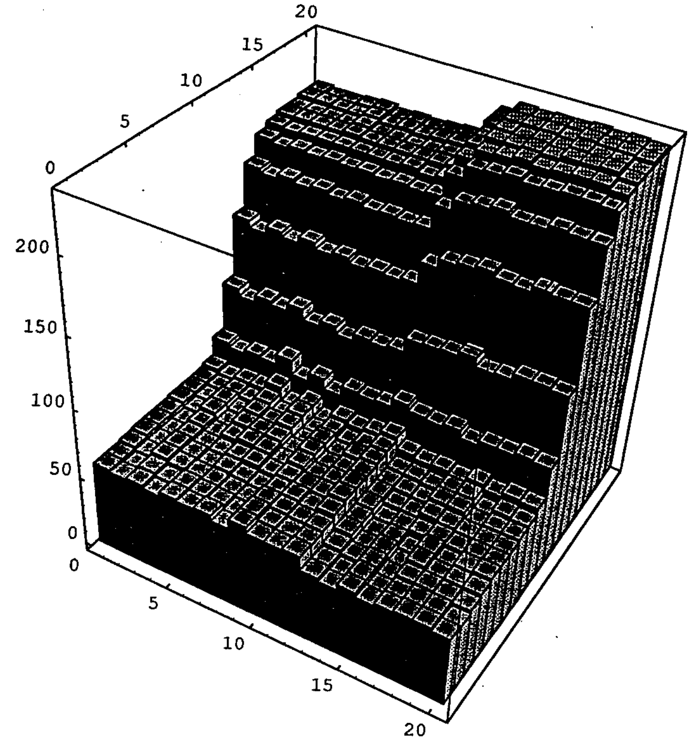


Figure 4-13. Barchart of Checkerboard Data of File s3

The contour plot of figure 4-14 illustrates that the gray scale gradient shifts slightly to the right in the bottom half of the checkerboard, an artifact of the analog video signal response. This is important to note in the following Gabor phase analysis, which picks up this subtle shift. Let us now compute the position of the edges in the raw data by interpolation as we did for the data of file s0. Interpolating raw values at the top in row 7 of figure 4-10 using 63 for a low and 203 for a high pixel value in the left and right patches, yields column 12.54 as the subpixel address for the mean value of 133 using equation 4-1. Doing the same for the lower half at line 18 (low Of 54, high of 230) yields 12.31 as the subpixel address for the mean value of 142.

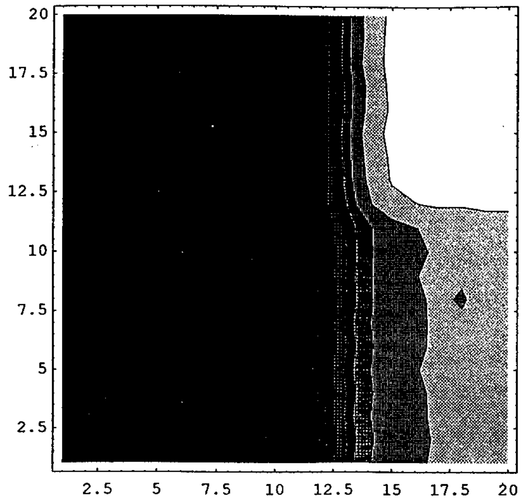


Figure 4-14. Contour Plot of Checkerboard Data of File s3

Figure 4-15 illustrates the numerical data from file s3g, the Gabor phase of file s3. Figure 4-16 plotted as elevation and 4-17 as a contour plot. We seek the contour line for 90° and calculate the subpixel edge location it signifies by interpolation according to equation 4-2. After offsetting by 3.5 pixels for registration with the s3 raw data, we get 12.55 (compared to 12.54 for raw value interpolation) for the top edge and 12.3 (compared to 12.31 for raw value interpolation) for the lower. Note that these are within 1/100th of a pixel of the interpolated raw values!

0	0	0	0	130	128	121	109	91	72	60	54	52	46	0	0	0	0	0	0
0	0	0	0	130	129	121	109	91	72	60	54	52	45	0	0	0	0	0	0
0	0	0	0	131	129	121	109	91	72	60	54	52	45	0	0	0	0	0	0
0	0	0	0	131	129	121	109	91	72	60	54	52	46	0	0	0	0	0	0
0	0	0	0	132	129	121	109	91	72	60	54	52	44	0	0	0	0	0	0
0	0	0	0	132	128	121	109	90	72	60	54	53	0	0	0	0	0	0	0
0	0	0	0	131	128	120	108	90	72	60	54	53	0	0	0	0	0	0	0
0	0	0	0	130	127	119	107	89	71	60	54	53	0	0	0	0	0	0	0
0	0	0	125	130	127	119	107	88	70	60	54	53	0	0	0	0	129	130	48
0	0	0	123	129	126	119	106	87	70	59	54	52	0	0	0	0	131	132	47
0	0	0	122	129	126	119	106	87	70	59	54	52	0	0	0	0	132	133	47
0	0	0	122	129	126	119	106	86	69	59	54	51	0	0	0	0	132	133	47
0	0	0	122	129	127	119	106	86	69	59	53	51	0	0	0	0	133	133	47
0	0	0	124	129	127	119	106	86	69	59	53	51	0	0	0	0	129	129	47
0	0	0	125	130	127	118	105	86	69	59	53	51	0	0	0	0	124	123	47
0	0	0	125	129	126	118	105	86	69	59	53	51	0	0	0	0	118	0	47
0	0	0	126	129	126	118	105	86	69	59	54	52	0	0	0	0	0	0	0
0	0	0	124	128	125	118	105	86	69	59	54	52	46	0	0	0	0	0	0
0	0	0	122	127	125	118	105	86	69	59	54	52	52	0	0	0	0	0	0
0	0	0	122	127	125	118	105	86	69	59	54	52	54	0	0	0	0	0	0

Figure 4-15. Gabor Phase of Checkerboard Data of File s3

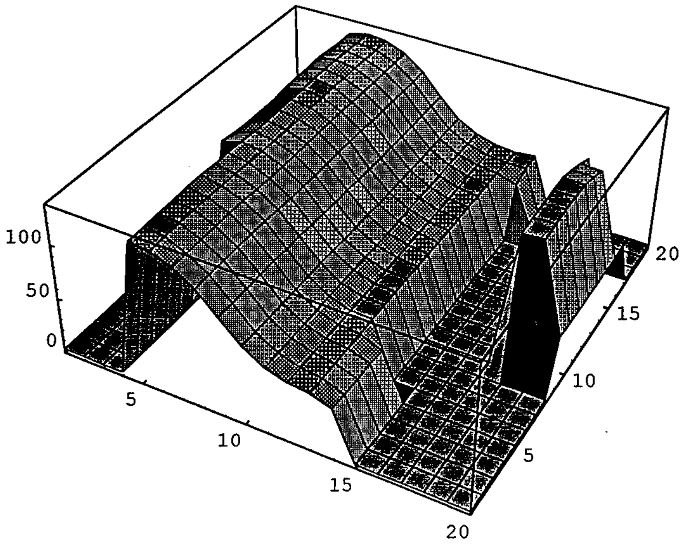


Figure 4-16. Elevation Plot of Gabor Phase of Checkerboard Data of File s3

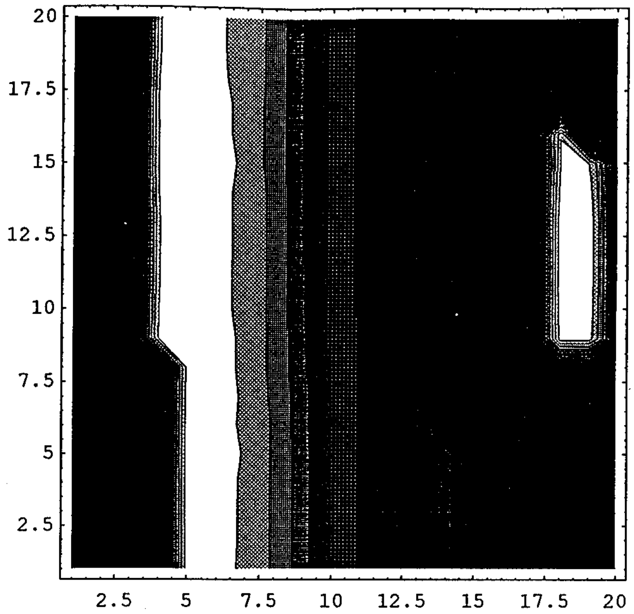


Figure 4-17. Contour Plot of Gabor Phase of Checkerboard Data of File s3

Scene 7: Subpixel Optic Flow (Disparity) Measurement via Gabor Phase

This is an animated scene illustrating a vertical edge moving in a regular sequence of subpixel increments. We mounted a linear motion stage at the position of the test pattern board shown in figure 4-1. Direction of motion was horizontal, parallel to the camera image plane. A vertical black/white edge pattern was mounted on the slide stage. The procedure consisted of capturing and image, storing it in file s70, moving the stage 1mm to the right, capturing an image into file s71, and repeating the motion and capture to the position 5mm to the right, which was stored in file s76. Each of these files contained data very much like that shown in figure 4-3, namely 20 rows by 20 columns of pixel data.

We then processed each of the six files for Gabor phase, yielding six output files s70g - s75g. From these we extracted the first row of 18 pixels from each, and merged them into "motion picture" file, s7ng, representing each frame of the motion as a row of pixel data. Figure 4-18 lists this data in numerical form.

```
1)  0   0   0 121 129 127 119 107  88  70  59  53  51  47   0   0   0   0
2)  0   0   0   0 126 127 122 112  96  76  63  55  51  49   0   0   0   0
3)  0   0   0   0 126 128 125 115 101  81  66  56  52  49   0   0   0   0
4)  0   0   0   0 126 129 126 118 105  86  69  59  53  51  47   0   0   0
5)  0   0   0   0   0 128 128 122 111  95  76  63  55  52  49   0   0   0
6)  0   0   0   0   0 126 129 125 115 101  81  66  57  52  50   0   0   0
```

Figure 4-18. Gabor Phase Record for Rightward Motion

Figure 4-19 depicts the data as a contour map, whose slanted curves represent the motion.

```
In[22]:=
    ListContourPlot[Table[list[[i]], {i,6}] ]
```

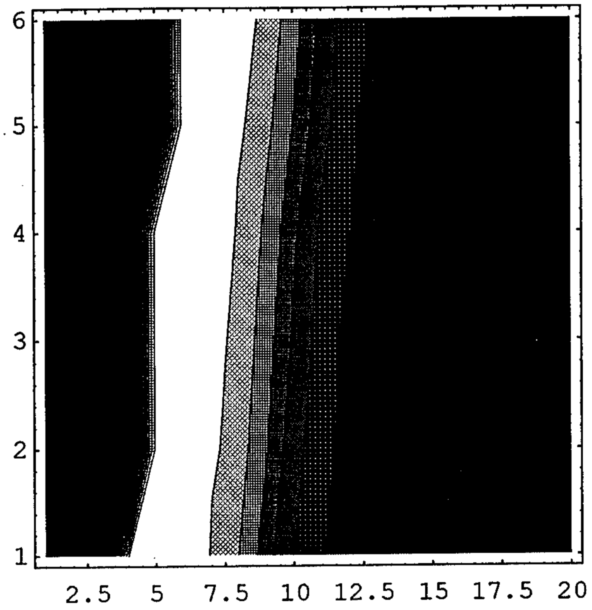


Figure 4-18. Contour Plot of Gabor Phase Record for Rightward Motion

We interpolated the 90^0 subpixel address according to equation 4-2 for each row, yielding the sequential edge position pixel addresses shown in table 4-3 below.

Line No.	1	2	3	4	5	6
Pixel Addr:	8.89	9.30	9.55	9.79	10.26	10.55

Table 4-3. Motion of Gabor Phase through 1mm Displacements

According to the calibration data collected on scene s0, shown in table 4-2, each 1mm motion corresponds to a .29 pixel excursion. The data above averages .33 pixels per motion, but varies from .24 to .53. The large variance may be accounted for in part by the manual procedure of moving the sliding stage eyeballed against a visual reticle. This could be corroborated by the reader by comparing the Gabor derived pixel address with the raw data derived pixel address, as we did for the checkerboard pattern. Regardless, the results are quite good, as shown by the nearly linear trend in progression of phase versus motion, illustrated in figure 4-19.

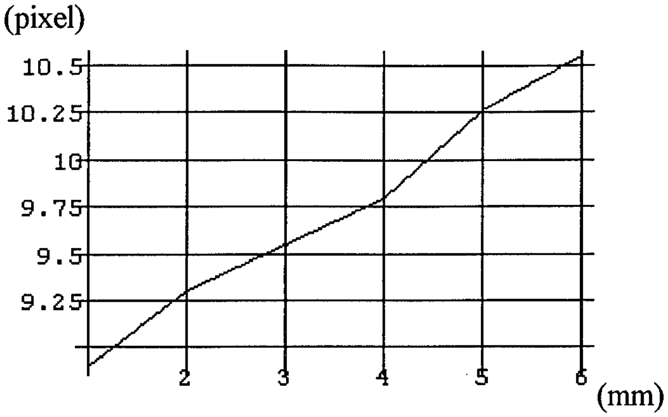


Figure 4-19. Gabor Phase as a Function of Displacement (Horizontal Axis)

Experimental Conclusions: Impact for Binocular Stereo

The analysis of Scene 3 (file s3), the checkboard pattern, indicated 1/100th subpixel accuracy for Gabor phase edge localization. In binocular vision, accuracy of angular resolution translates directly into range resolution accuracy. Consider two cameras symmetrically converged on a target. Figure 4-20 illustrates a top view of two corresponding pixels at the center of the field of view of each camera. The shaded quadrilateral in the center is a voxel expressing the position uncertainty of a 3-D point which is visible by both pixels simultaneously. We approximate this quadrilateral by a *parallelogram* to simplify the geometric analysis which follows, justifying the approximation by noting that for small angles, i.e. pixel subtense of a few milliradians, the error incurred is less than one percent; that is, pixel subtense is a differential quantity.

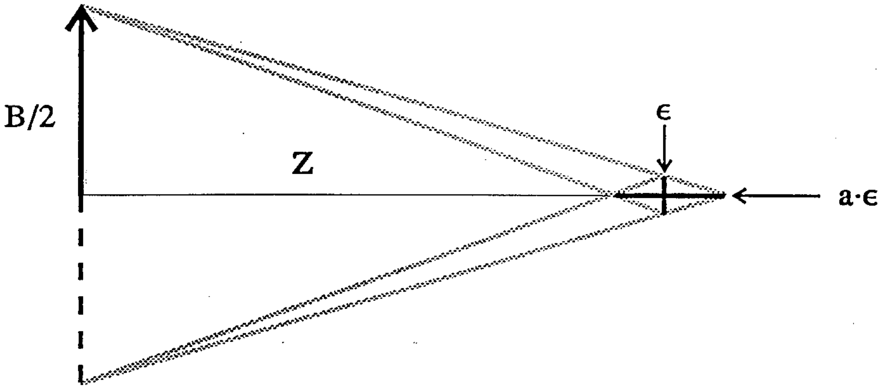


Figure 4-20. Voxel geometry at the binocular center

The key observation is that the aspect ratio of such a voxel is the ratio of range Z to half-baseline, namely,

$$a = \frac{2Z}{B} \ . \tag{4-3}$$

as seen by similar triangles in figure 4-20. The length of the voxel is the magnitude of the range error (i.e. range accuracy) for binocular stereo. Since this quantity is proportional to pixel subtense, we can quantify the Gabor phase advantage directly in terms of pixel size as follows. As a practical example, consider a binocular camera system based on 512 pixel diameter image planes with 90° fields of view. For a baseline B of 25 centimeters, and a vergence distance of 2 meters, equation 4-3 tells us that voxel aspect ratio is 16-to-1. Pixel cross section is simply range times angular subtense in radians,

$$\frac{\pi}{2 \cdot 512} \times 2 \ meters = 6 \, mm \tag{4-4}$$

whence range uncertainty is

$$16 \times 6 \, mm = 9.6 \, cm \ . \tag{4-5}$$

Now, improvement by a factor of 100-to-1, verified by the experiments on image s3, slices the voxel into 100-by-100 sub-parallelograms of the same aspect ratio as the parent. Hence the range accuracy is .01 times the quantity above, or

$$9.6 / 100 \ cm = .96 \, mm \ . \tag{4-6}$$

Thus, binocular stereo accuracy is refined to better than 1mm for this example.

4.2 Character Recognition

We applied our discrete Gabor filters to the problem of handwritten character recognition. A neural network was constructed to serve as a classifier. The database included in our study contained 17 forms which were written by 17 distinct persons. Each form had ten classes (characters 0 - 9), and each class had 12 characters. Figure 1 illustrates such a form *(Note: figure and table numbering is local to this section)*. There were a total of 2040 characters in the database. Two thirds of the characters were used for training; and one third for testing. Figure 2 shows an example of the characters included in our testing database. The matrix size of each character was 64 x 64 pixels, 8 bits/pixel.

Each character was decomposed into four levels of scale in a pyramid of discrete Gabor filters. The neural network was driven by coefficients of the coarsest level. Figure 3 shows four levels of sine components. The four blocks shown in the upper left are the four directional coarsest components (thus the size of each block was 4 x 4). We selected a size of 8x8 for representation at the inputs of the neural network. The number of input nodes of the neural network was 64 and the number of output nodes 10. Cosine components of the corresponding decomposition are shown in Figure 4.

In our experiments, we used 8 x 8 sine components as inputs of the neural network. The number of hidden nodes was 3 (chosen experimentally). In our first attempt, the network did not converge for all our training samples. A similar result was obtained for cosine components. We then tried to include both components and the dc component to form a larger block (8 x 16). We replaced one of the sine components with a dc component as shown in Figure 5. This representation exhibited faster convergence speed but still was not able to converge. A filter size (8x8) may be one for failure reason. This was the same size as the coefficients of the coarsest level before down sampling. We used mirror reflected extension at the border for convolution.

Choosing a bigger block was our next consideration. In order to avoid complicated computations, we first employed 16 x 16 cosine components and replaced one of the blocks with dc components. This is shown in Figure 6. The convergence speed was improved and the number of wrong classifications was reduced and finally this network did converge. We observed 223 errors out of 880 testing characters. The number of errors observed for network training converged to 0 after 685 iterations. The running time was about seven days on a SUN SparcStation Model 10/30.

The confusion matrix for training is shown in table 1, illustrating that there were no errors in recognizing the characters for which the network was trained, that is, character *i* was always recognized as character *i* and character *j* was never misinterpreted as character *i*.

	1	2	3	4	5	6	7	8	9	10
1	136	0	0	0	0	0	0	0	0	0
2	0	136	0	0	0	0	0	0	0	0
3	0	0	136	0	0	0	0	0	0	0
4	0	0	0	136	0	0	0	0	0	0
5	0	0	0	0	136	0	0	0	0	0
6	0	0	0	0	0	136	0	0	0	0
7	0	0	0	0	0	0	136	0	0	0
8	0	0	0	0	0	0	0	136	0	0
9	0	0	0	0	0	0	0	0	136	0
10	0	0	0	0	0	0	0	0	0	136

Table 1. Confusion Matrix for Training Set

The confusion matrix for testing, shown in table 2, tells a different story. This indicates data for characters for which the network was not trained. The large values of off-diagonal elements indicate rampant confusion between character *i* and character *j* throughout. How could the two be so different? The numbers, plus the very long time to convergence (7 days!) indicates that the network *memorized* the training characters rather than *generalizing* them. This is not good. Why did this discrete Gabor filter perform so much more poorly than wavelets of other kinds? One possible answer is that most 2-D wavelets include cross-terms sensitive to junction shape (X's, Y's, T's etc.) corresponding to the second order crossed axis partial derivatives. Our discrete Gabor filter only detects same-axis edges, a directional derivative-like operation. Thus, the elements of the computational basis as presented by the discrete Gabor filter need to be augmented by higher order and cross-term derivatives.

	1	2	3	4	5	6	7	8	9	10
1	8	3	4	5	8	8	7	6	11	8
2	2	11	10	5	9	4	13	3	5	6
3	7	3	6	12	9	5	11	7	3	5
4	2	5	7	9	2	10	10	9	8	6
5	4	4	5	11	9	7	6	8	6	8
6	7	5	10	10	4	4	7	3	9	9
7	7	10	5	3	7	11	12	5	6	2
8	12	2	5	9	9	4	8	12	1	6
9	8	6	5	7	5	8	7	6	8	8
10	7	3	5	7	10	8	8	7	8	5

Table 1. Confusion Matrix for Testing Set

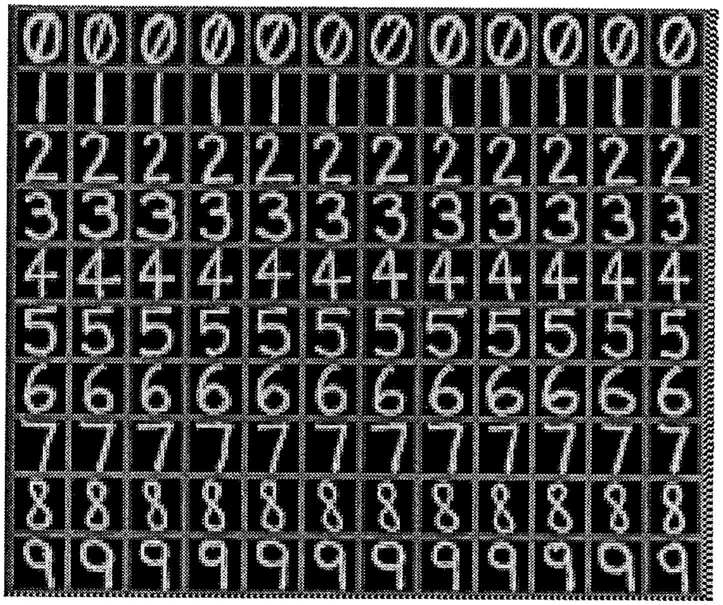


Figure 1: A form of database.

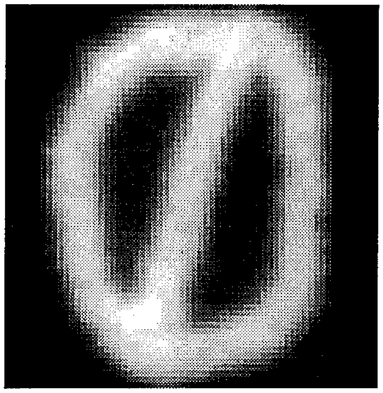


Figure 2: A sample character from database.

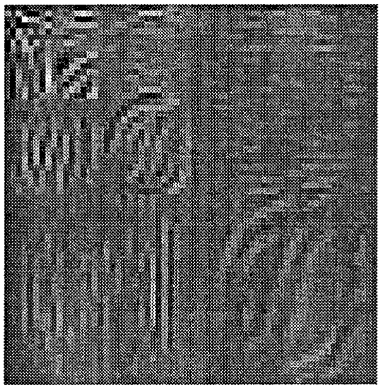


Figure 3: Four levels of analysis for sine components.

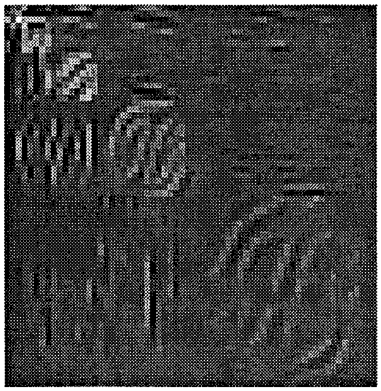


Figure 4: Four levels of analysis for cosine components.

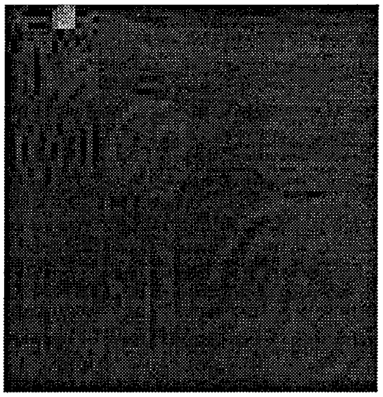


Figure 5: Four levels of analysis for sine, cosine and dc components (*log* scale).

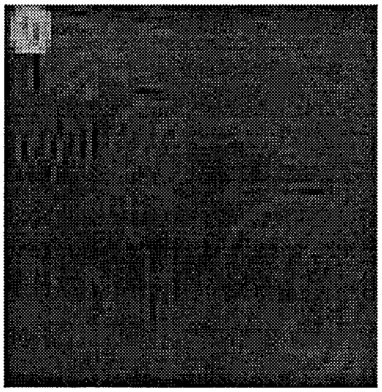


Figure 6: Three levels of analysis for cosine and dc components (*log* scale).

5. CONCLUSIONS

This research analyzed the theoretical characteristics of the discrete Gabor filter and tested the filter on real imagery. Several unexpected results emerged. On the theoretical front, we found that the discrete Gabor filter, as formulated by these investigators, is not a wavelet, and is not a complete nor orthogonal image decomposition function. Thus image compression and reconstruction cannot be done solely with this set of functions. The discrete Gabor filter strongly resembles a second order directional derivative. What is missing for completeness in image decomposition is the zeroth, first, and mixed second order derivatives. Thus, by augmenting the set proposed here with such additional filters, a complete set might be constructable, but was not done in this study.

On the applications side, we found that the Gabor filter performed superbly to 1/100th pixel accuracy for binocular disparity measurements. However, there was an unexpectedly large phase spread across visual contrast boundaries, very likely due to the poor bandpass of analog video signals feeding the frame buffer. This might be corrected by using a digital camera instead of the usual video CCD.

In the recognition of handwritten characters, the discrete Gabor filter performed more poorly than wavelets, very likely because of the lack of second order mixed partial derivative type components which would have caught junction topologies of characters with “X” and “Y” subgeometries. Thus, this failure is related to the lack of completeness discovered in the theoretical analysis.

The negative results above precluded the design of a general purpose “silicon cortex” architecture, because the components do not possess general computation power. Nevertheless, follow-on study would augment the elementary components proposed here, testing them on a high performance general purpose processor such as the C80 system from General Imaging Corporation, or the CNAPS from Adaptive Solutions. A digital camera input is essential.

Despite some negative results, there are also bright spots. The high performance of phase disparity measurement, and its low computational cost, suggest incorporation into real-time vergence servos for binocular stereo camera platforms. TRC manufactures such hardware and could well incorporate the vergence software for enhanced performance.

6. REFERENCES

[1] M. J. Bastiaans, "Gabor's Expansion of a signal into Gaussian elementary signals," *Proc. IEEE*, vol. 68, no. 4, pp. 538-539, April 1980.

[2] M. J. Bastiaans, "A sampling theorem for the complex spectrogram, and Gabor's expansion of a signal in Gaussian elementary signals," *Opt. Eng.*, vol. 20, no. 4, pp. 594-598, July 1981.

[3] A. Bovik, M. Clark, and W. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 55-73, Jan. 1990.

[4] P. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Comm.*, vol. 31, no. 4, pp. 532-540, April 1983.

[5] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Reg. Conf. Ser. Appl. Math. 61, Philadelphia: Soc. Ind. Appl. Math, 1992.

[6] J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vis. Res.*, vol. 20, pp. 847-856, 1980.

[7] J. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1169-1179, July 1988.

[8] F. Farrokhnia and A. K. Jain, "A multi-channel filtering approach to texture segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition*, Maui, Hawaii, pp. 364-370, June 1991.

[9] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *J. Opt. Soc. Am. A*, vol. 4, no. 12, pp. 2379-2394, Dec. 1987.

[10] B. Friedlander and B. Porat, "Detection of transient signals by the Gabor representation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 2, pp. 169-180, Feb. 1989.

[11] D. Gabor, "Theory of communication," *J. Inst. Elect. Engr.*, vol. 93, pp. 429-457, 1946.

[12] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167-1186, 1991.

[13] R. Mehrotra, K. R. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognition*, vol. 25, no. 12, pp. 1479-1494, 1992.

[14] R. S. Orr, "The order of computation for finite discrete Gabor transforms," *IEEE Trans. Signal Processing*, vol. 41, no. 1, pp. 122-30, Jan. 1993.

[15] M. Porat and Y. Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 452-468, July 1988.

[16] M. Porat and Y. Y. Zeevi, "Localized texture processing in vision: analysis and synthesis in the Gaborian space," *IEEE Trans. Biomedical Eng.*, vol. 36, no. 1, pp. 115-129, Jan. 1989.

[17] S. Qian and D. Chen, "Discrete Gabor transform," *IEEE Trans. Signal Processing*, vol. 41, no. 7, pp. 2429-2438, July 1993.

[18] C. F. R. Weiman, "Efficient discrete Gabor functions for robot vision," *Proc. SPIE Conf. on Wavelet Applications*, vol. 2242, pp. 148-160, 1994.

[19] J. Yao, "Complete Gabor transformation for signal representation," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 152-159, April 1993.